絵でわかる たのしいMSXパソコン入門

関口泰夫
新星出版社

新星出版社

●はじめにいっておきたいこと

ＭＳＸタイプのコンピュータの人気は、やっぱりすごいもののようです。小学生の人もおとなの人も、いちどＭＳＸに手を触れてしまえば、すっかりとりこになってしまうのでしょう。

ほかのタイプのパソコンとくらべてみても、機能もおとるところはないようですし、それでいて値だんは安いというのですから、人気のほどもうなずけますね。それに、なんといっても、ＭＳＸBASICという共通のことばを持っているのがいいのです。ＭＳＸ用のソフトウェアなら、どれでも自分のマシンで使えるのですから、パソコン入門用のマシンとして、いちばん適しているといってもいいのではないでしょうか。

ＭＳＸは、それほどすばらしいマシンなのです。これを遊ばせておく手はないのです。うんと使いまくってやりましょう。

ただ、パソコンにはいろいろ約束ごとがあったり、BASICという英語みたいなものがあったりして、めんどうくさいなあ、なんて思わないでください。また、BASICをマスターしてからキーをたたこうなんて、決して思わないでください。

はじめは、なにもわからなくてもいいのです。いわれたとおりにキーをたたくことです。よくわからないページがあったら、とばして先に進むのもいいでしょう。おぼえようとしても、なかなかおぼえられないのがコンピュータですし、おぼえようとしなくとも、キーをたたいているうちに、自然にわかってしまうのもコンピュータなのです。とにかく、やってみることをおすすめします。

（著者しるす）

もくじ

パート 0 コンピュータってな〜に？

もくじ

パート3　やさしいプログラミング

パート0

コンピュータってな～に？

1 こんにちはパソコン

ゲーム機としても楽しいけれど

いまはもう、パソコンを全然知らない人なんて、ほとんどいないはずだ。でも、パソコンっていうことばから、なにを思い浮かべるかは、みんな違っているかもしれない。

色あざやかなパソコンゲームやテレビゲームの画面を思い浮かべる人、むずかしいさんすうの計算式を連想する人、文章をどんどん印字していくワープロのようなものを想像する人と、いろいろあるはずだ。

ホームコンピュータ級の小型パソコンでも、使い方は数かぎりないといっていいくらいあるのだ。

楽しく遊ぶホビーの用途だけでも、ゲームのほかに、ゲームソフトやレコード、ビデオテープなどのデータバンクや、アニメーション、音響、カラーグラフィックスなど、いくらでもあるのがパソコンの世界といっていいだろう。

もちろんホビーだけじゃないよ。会社の事務だってこなすし、勉強の手伝いもＯＫという万能マシンなのだと考えておこうよ。

パソコンの使われ方いろいろ

家庭では

ゲームするだけが能じゃない。ダイエットメニューを作ったり電話帳を作ったり。

学校や塾では

教育用の利用範囲はどんどん大きくなる。各課目の辞書としての役割や得点計算、順位計算などお手のものだ。

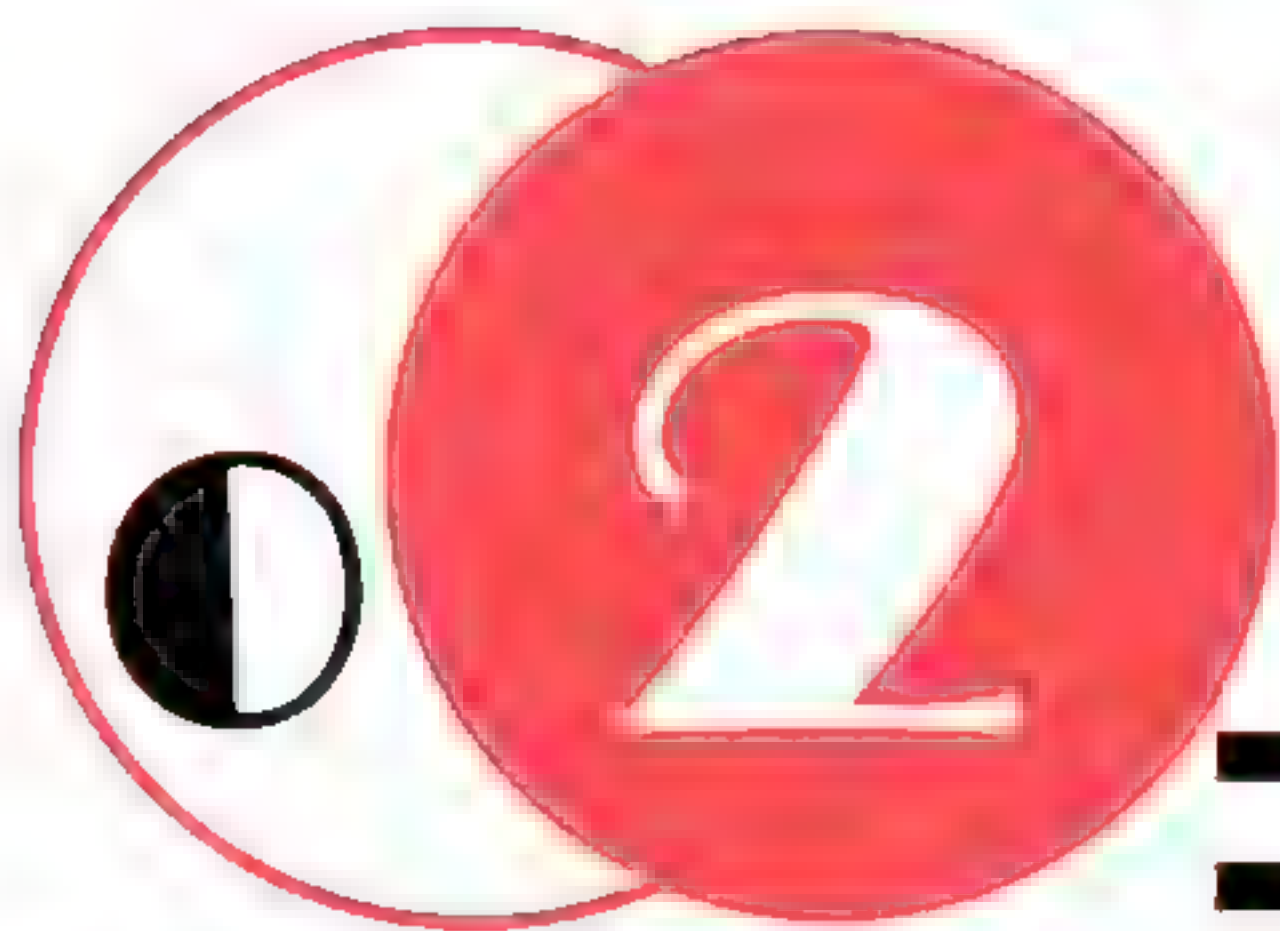
オフィスでは

営業や事務管理だけでなく、端末としてもデータバンクとしてもオフィスオートメーションには欠かせないマシンになっている。

パソコン応用分野の一部

(ホームコンピュータ)ゲームマシン／ホビー用データバンク／オーディオ・ビデオ制御／家計管理／食事献立表／家電制御／その他

(教育補助マシン) 各課目ごとの辞書／視覚教育教材／理数計算補助機／教育用シミュレータ／教材データバンク／その他

(オフィスオートメーション) ワードプロセッサ／事務・営業管理／事務データバンク／一般事務機／端末機／その他

2 家の中にはコンピュータだらけ

目には見えないけどはたらきつづけ

タイプライターみたいなキーボードと、テレビの画面そっくりなディスプレイと、紙に文字を印刷するプリンタがそろっていれば、だれだって、「ははあ〜ん、これがコンピュータだな！」ってわかっちゃう。

だけど、ちょっと見たところ、コンピュータではないのに、ほんとうは内部にコンピュータが入っているものは、台所にもダイニングルームにも、勉強部屋にも、たくさんころがっているんだよ。

たとえば、おかあさんがだいじにしているミシン。花柄でもほかの模様でも、どんどんししゅうしてしまうのも、中に入っているコンピュータがやらせているのだ。

ほかにもいろいろある。かってに走ったり曲がったり止まったりするプラモの戦車。暑かったら空気を冷やし、寒くなったらあたためる部屋の中のエアーコンディショナー、洗たく機などなど。みんなコンピュータが命令して動かしているものが多いのだ。

部屋の中をもう一度
見わたしてみよう
コンピュータが
入っている製品は
たくさんごろごろ
ころがっているよ

頭の中に
パソコンを入れれば
さんすうの宿題なんか
イチコロだ

身のまわりにあるコンピュータ

ＥＥカメラ

レンズを向けてシャッターを押すだけでいいＥＥカメラ。距離をはかったり、明るさを調べたりして、シャッター速度やしぼりを決める。

ミシン

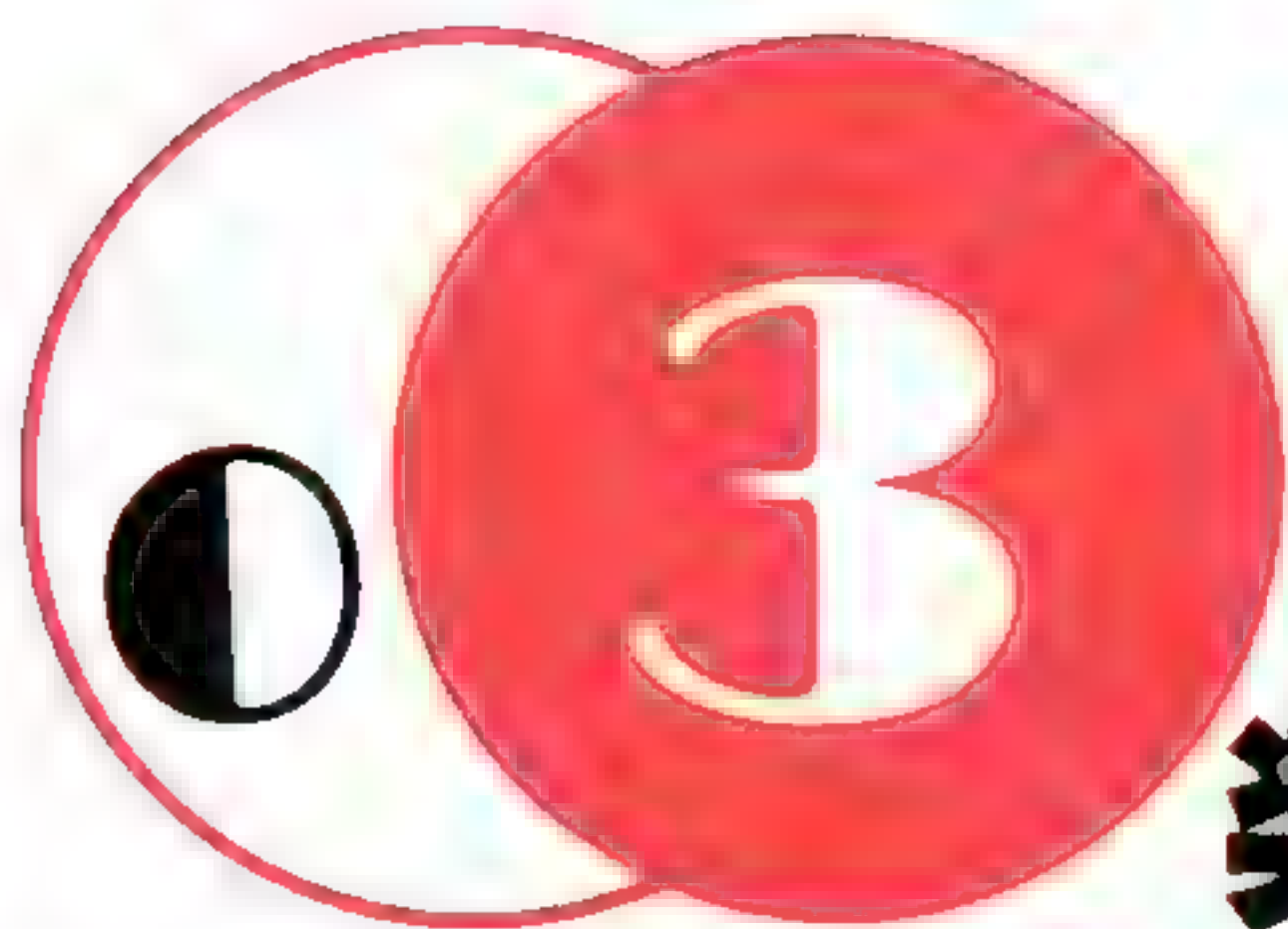
花柄やいろいろな模様を自由自在にししゅうしたりするのは、みんな内蔵コンピュータのはたらき。

クーラー

クーラーの自動温度調節もマイコンがやっているのだ。

自走プラモデル

たとえば戦車やジープなどのプラモデル。どんなスピードで、どこで曲がるかなど、みんな内蔵コンピュータしだいだ。

それだけじゃないよ
まわりを見わたしてごらん
洗たく機だって
そうじ機だって
新しい電化製品や
自動化機器には
ほとんど
マイコンが入っている

楽しいディスプレイ

しごとの結果が目で見てわかる

パソコンにはディスプレイがつきものだ。ディスプレイはテレビのブラウン管と同じで、そこに文字を書いたり絵を描いたり、いろいろなものを表示する。

パソコンのできるしごとはたくさんあって、ひと口で説明できないけど、ディスプレイに表示するのも、たいせつなしごとのひとつなのだ。ディスプレイがなかったら、パソコンなんて、おもしろくもおかしくもないってことになりそうだ。

パソコンがはたらいた結果を表示して見せるのがディスプレイだといっておこう。ディスプレイがあるから、しごとの結果がわかるのだ。

ディスプレイが、パソコンにとって、なくてはならないものだということがよくわかったはずだ。

ディスプレイの表示のしかたにも、いろいろあるよ。パソコンに命令するプログラムを表示させたり、表やグラフはもちろん、ゲームのキャラクタや、アニメーション、直線や曲線を組み合わせたグラフィックスなど、なんでも表示してしまう。

色は15色を使って
表示できるから
画面はカラフル
色いっぱいだよ

ディスプレイは
専用ディスプレイの
ほかに
家庭用テレビも
使えるのだ

パソコン画面表示のいろいろ

入力したプログラムの表示。まちがいがないかどうかなどをたしかめられる。

表示画面には
グラフィック画面と
テキスト画面があるよ

文字を表示するのが
テキスト画面なんだね

円グラフや棒グラフも自由自在。しごとにも勉強にも活用できる。

なんといっても楽しいのが、パソコンゲームの画面だ。ちょこまか動くキャラクタが、またかわいいよ。

カラフルに、ダイナミックに、そうぞうしく作ってみよう。

カラーグラフィックスは、手でデザインするより、パソコンにやらせたほうが、思いもしなかった効果が出たりして、楽しい。

パソコンって どんなマシンなの？

キーボードやテレビやテープや

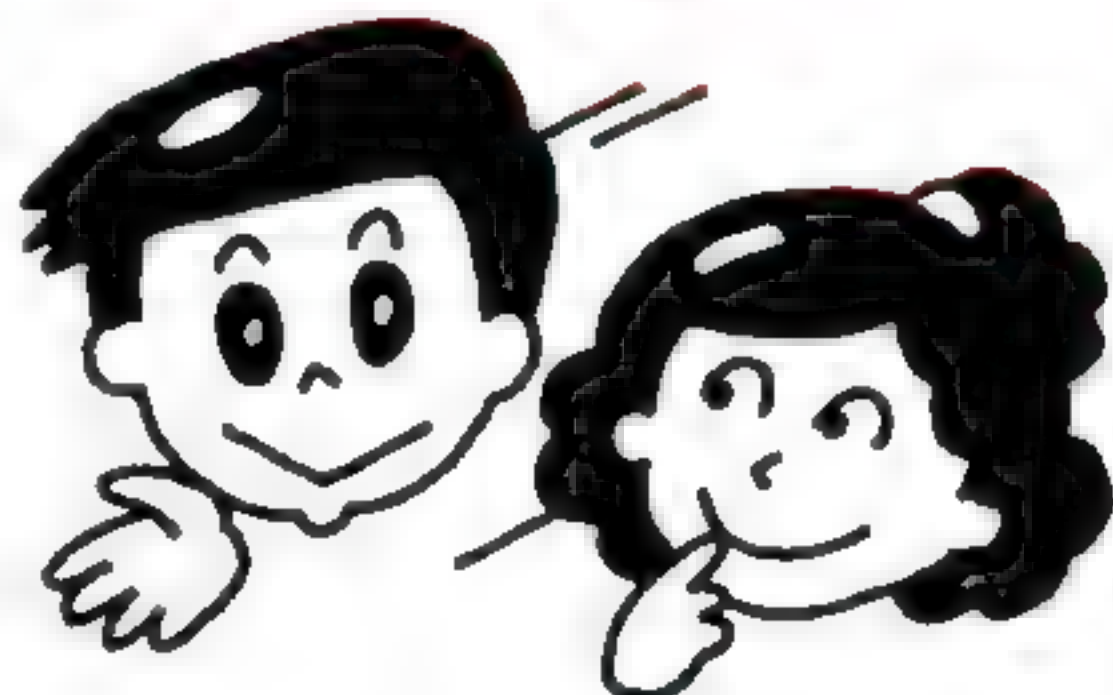
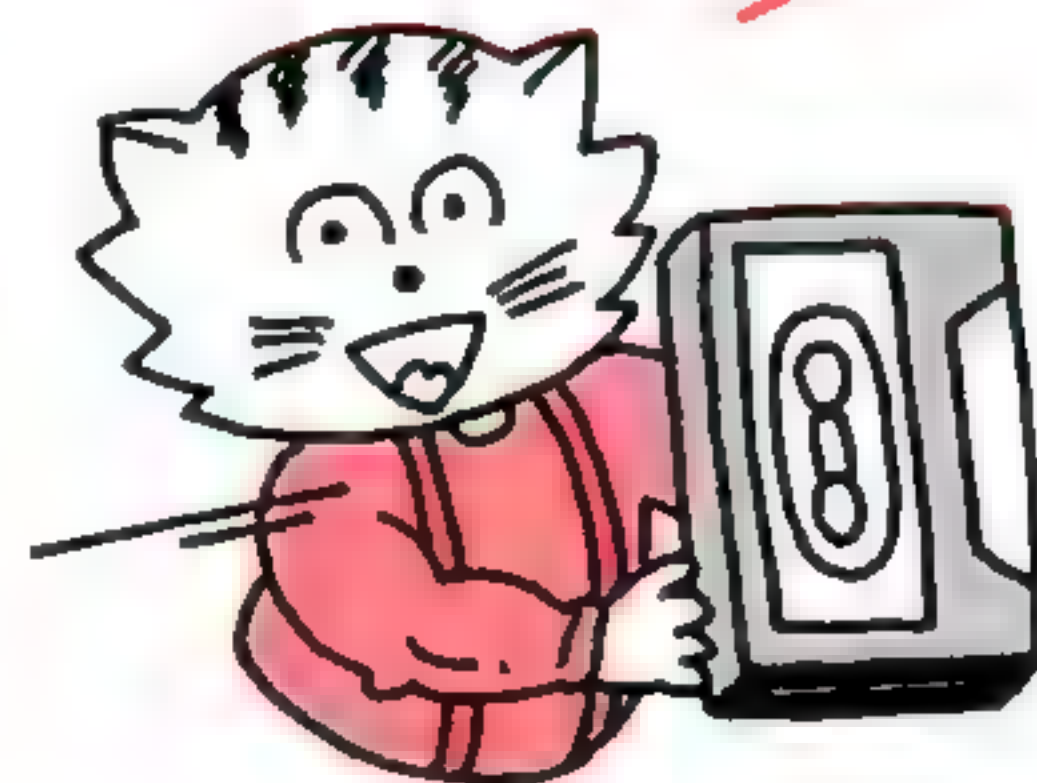
パソコンは、どんな機器があれば、パソコンといえるのだろう。

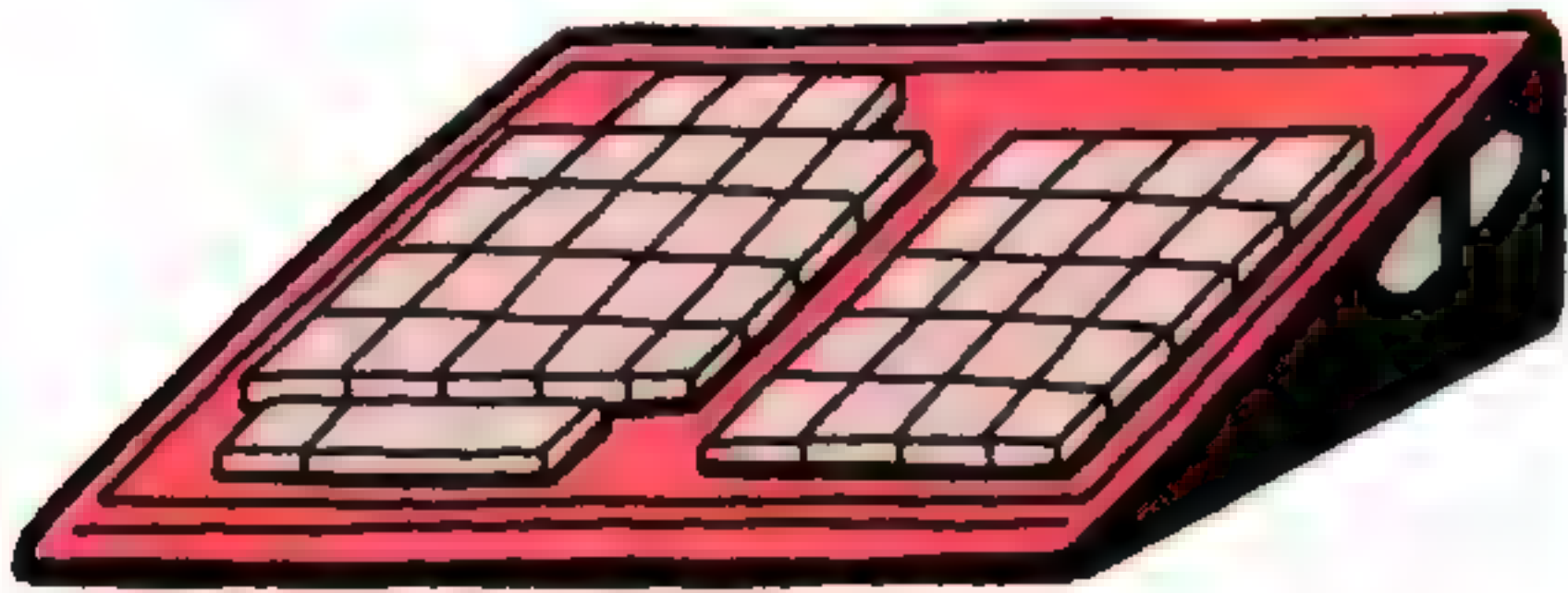
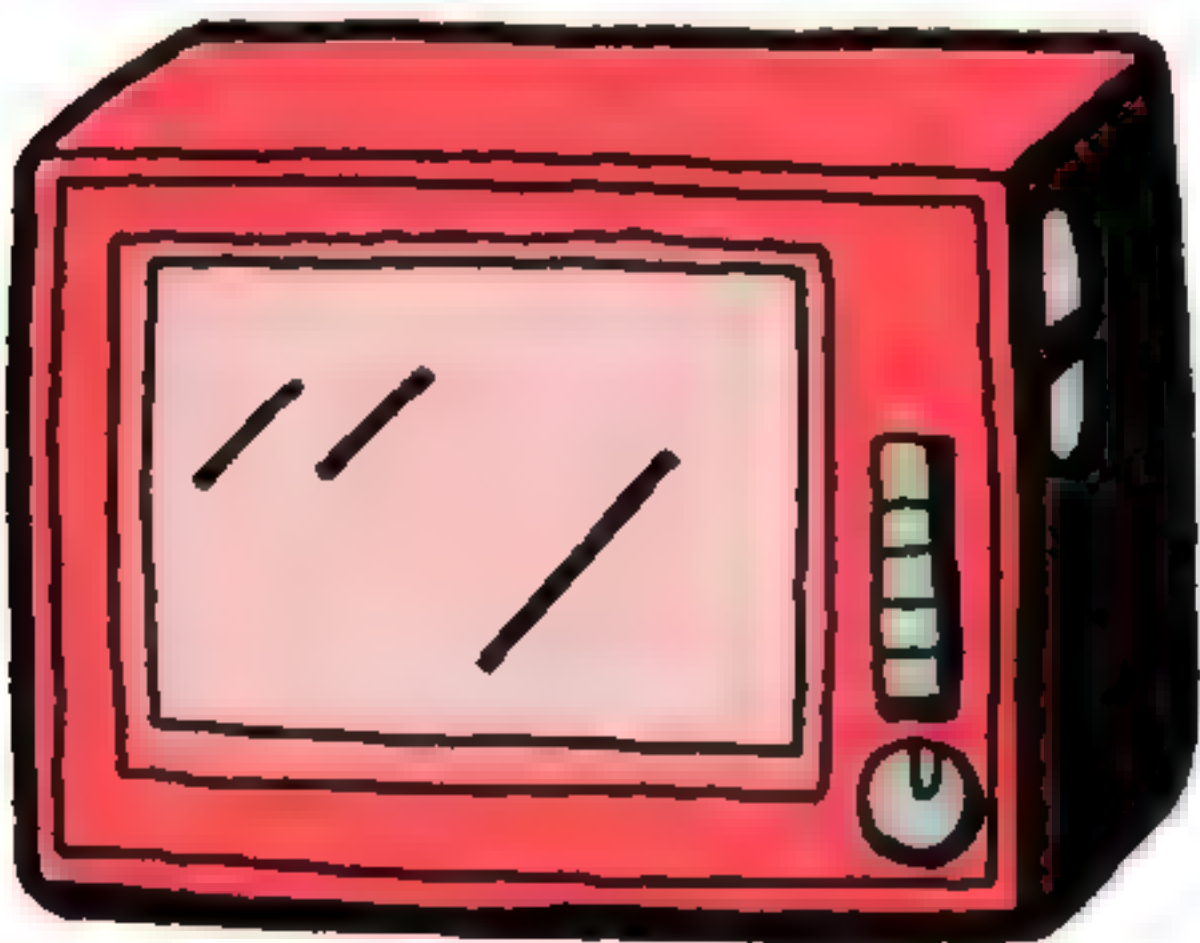
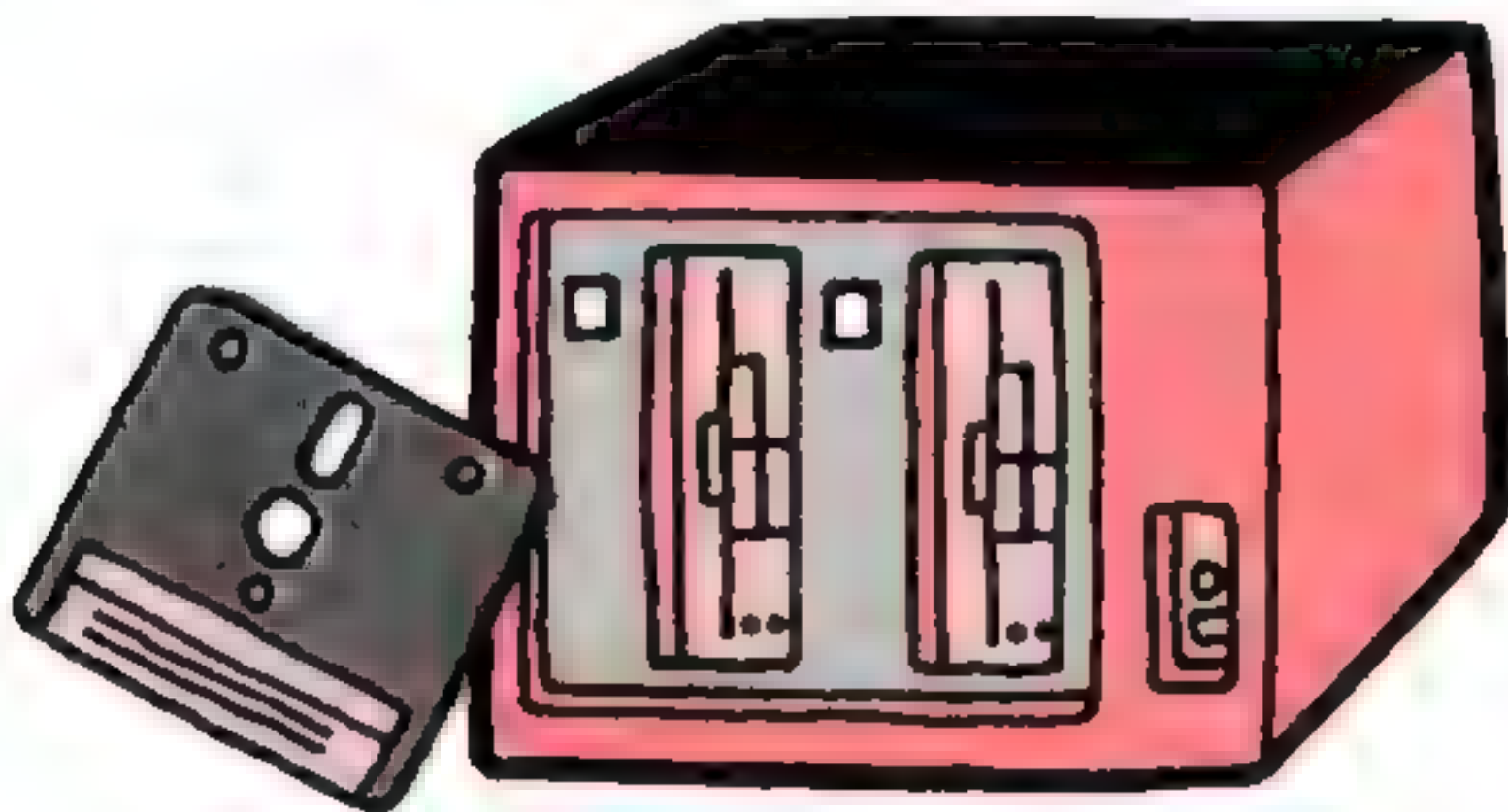
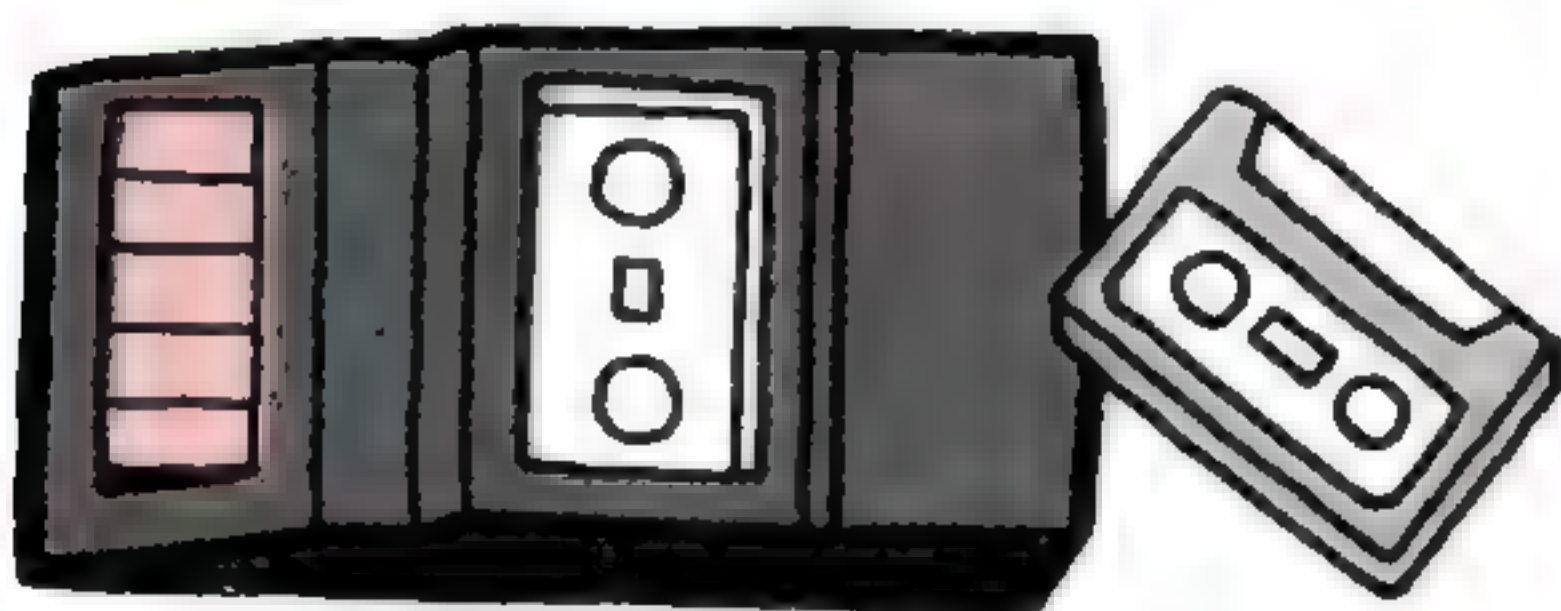
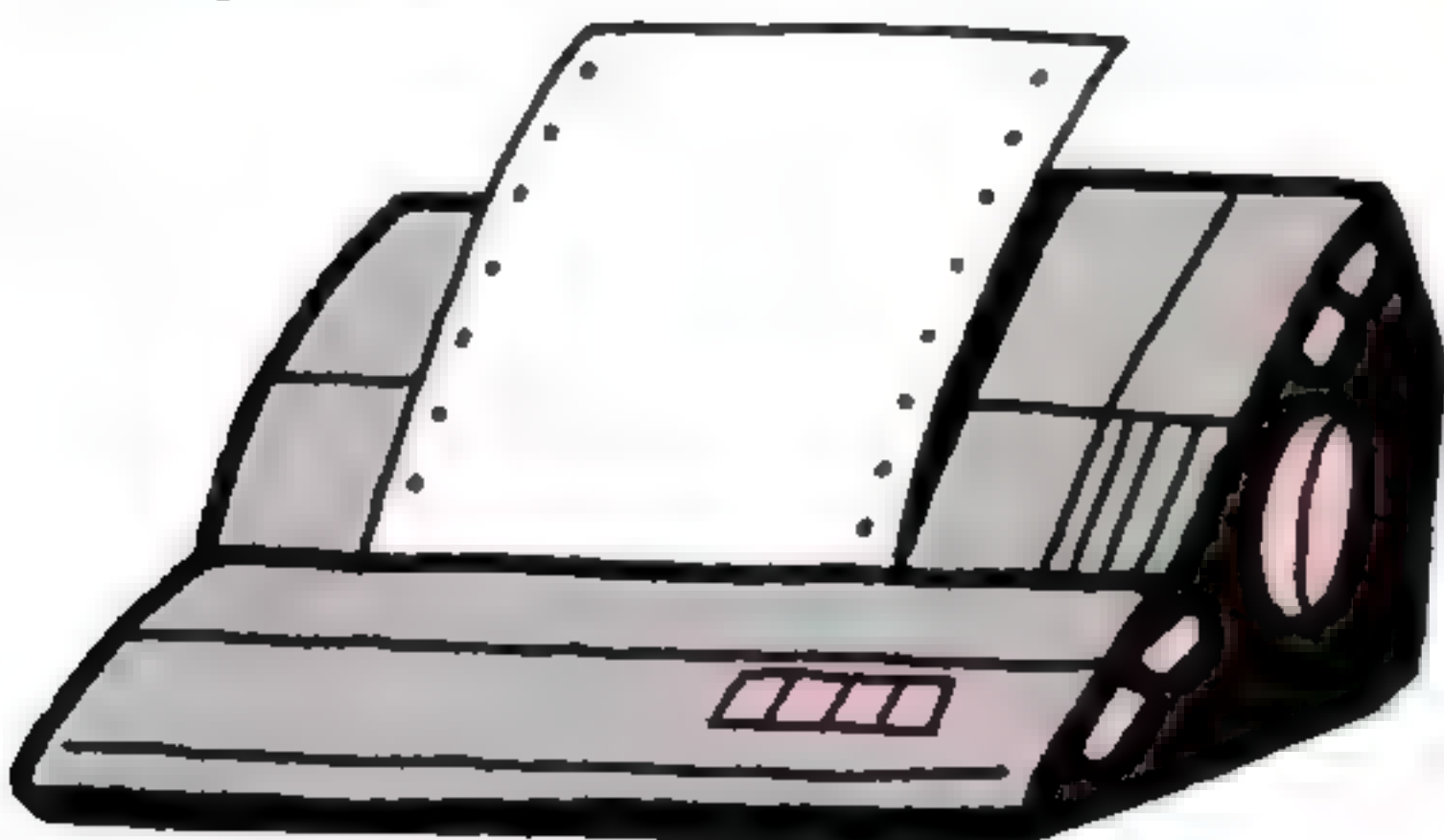
まず、本体。命令を記憶したり、命令を実行したりするパソコンの頭脳だ。タイプライターみたいに、文字や記号を打ち込むキーボードが付いているからすぐわかる。本体・キーボードともいう。

本体・キーボードがあれば、命令を入力できるし、それを記憶したり計算して実行したりできるけど、なにかまだものたりない。

実行した結果を知るためには、それを表示するテレビやディスプレイがほしい。結果を長く保存するには、紙に印刷しておくためのプリンタも必要だ。

こうした本体・キーボード以外の付属機器を周辺機器といっている。このほか、周辺機器の主なものには、カセットテープ、テープレコーダ、ディスク装置とディスケット。ほかにもまだまだあるが、これだけそろっていれば、まあ、いちにんまえのパソコンといっていいだろう。

テープは
音楽を聞く
ふつうのテープで
いいんだよ

音楽を録音するように
テープに
プログラムという
命令を
記憶させておくのだ

●パソコンの構成品

本体・キーボード

テレビ（ディスプレイ）

フロッピーディスク

テープレコーダ

プリンタ

本体・キーボード

キーボードは文字や記号などを本体に入力する入力装置。本体はここから入力された命令（プログラム）の記憶と実行をおこなう。

テレビ（ディスプレイ）

入力したプログラムを表示したり、本体が実行した計算結果や絵や図などを表示させる。出力装置のひとつ。

フロッピーディスク

一度作ったプログラムを記憶させておき、必要なときにここから呼び出して使用する。外部記憶装置という。

カセットテープとテープレコーダ

外部記憶装置のひとつで、はたらきはフロッピーディスクと同じ。手軽だがスピードがやや遅い。

プリンタ

プログラムや本体が実行した結果を保存するために、紙に印刷する。出力装置のひとつ。

パソコンのしごと

入力、記憶、演算、制御、出力

コンピュータにはいったいなにができるのだろう。コンピュータのしごとについて考えてみよう。

与えられたプログラムとデータにしたがって、しごとをするわけだが、しごとの内容は大きく分けて５種類ある。

入力、記憶、演算、制御、出力が、コンピュータのしごとだ。大きなコンピュータも、パソコンぐらいの小型のコンピュータも、できるしごとの内容は同じだ。

プログラムは
コンピュータに与える
しごとの内容を書いた
命令書なのだ

パソコンのしごとの種類

入力……プログラムやデータを外部から本体の記憶装置に入れるしごと。キーボードやジョイスティックが入力装置だ。

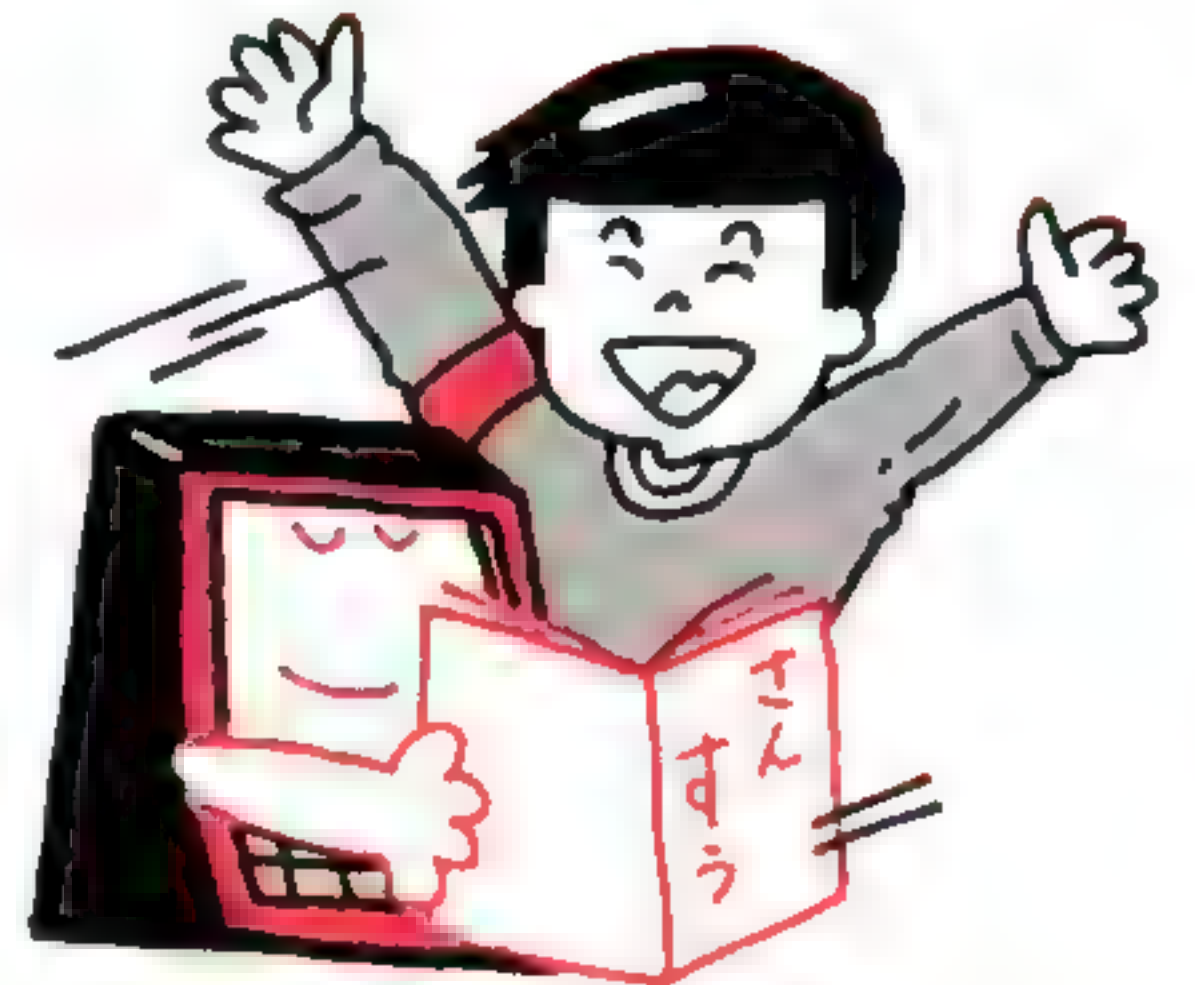
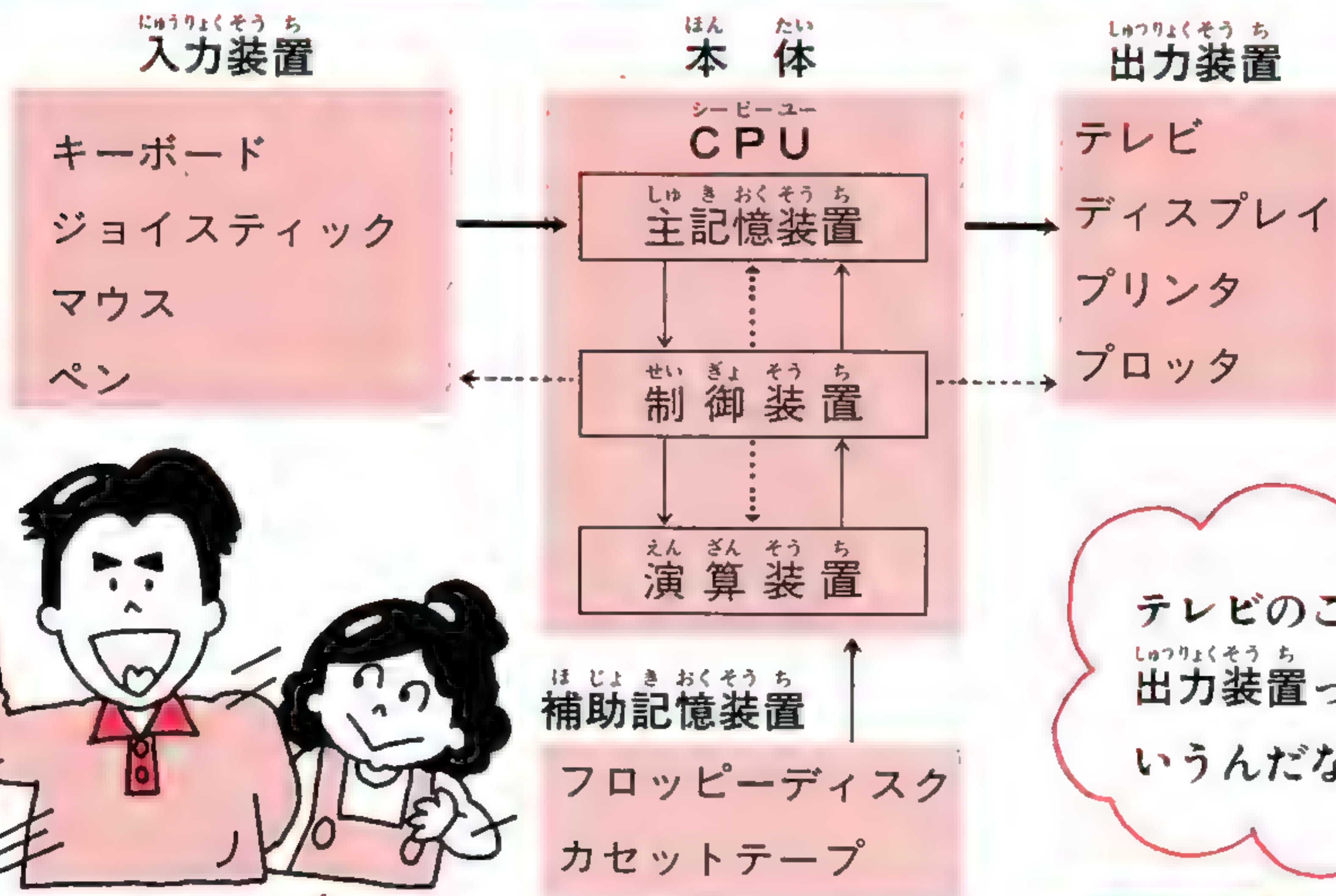
記憶……入力したプログラムやデータを記憶しておくしごと。本体内部の記憶装置のほかに、フロッピーディスクやカセットテープなどの外部記憶装置がある。

演算……記憶装置に記憶したデータを使い、プログラムの命令どおりに、計算や処理

本もののコンピュータは
５つの要素を
そなえていれば
いいんだな

の実行をするしごと。本体内部にある中央演算処理装置（ＣＰＵ）がおこなう。

制御……コンピュータを組み立てているすべての装置が、それぞれしごとの範囲を守って正しく動作するようにコントロールするしごと。

出力……演算装置の実行結果を外部に出力して知らせるしごと。テレビ（ディスプレイ）やプリンタが出力装置である。

●パソコンのシステム

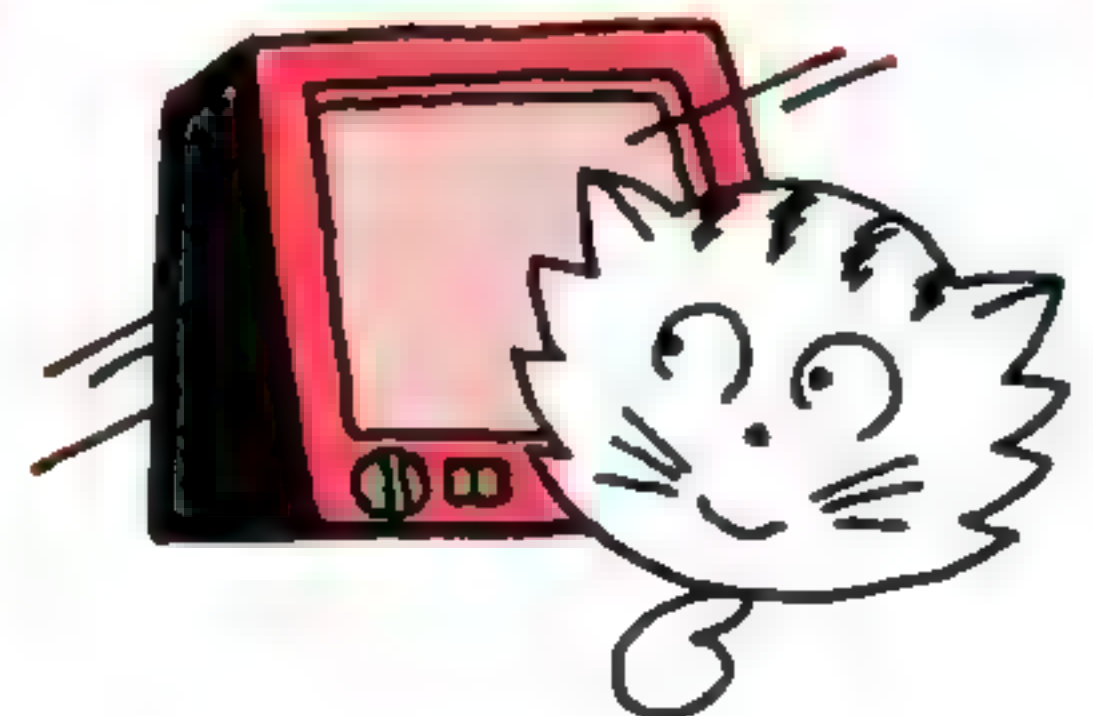
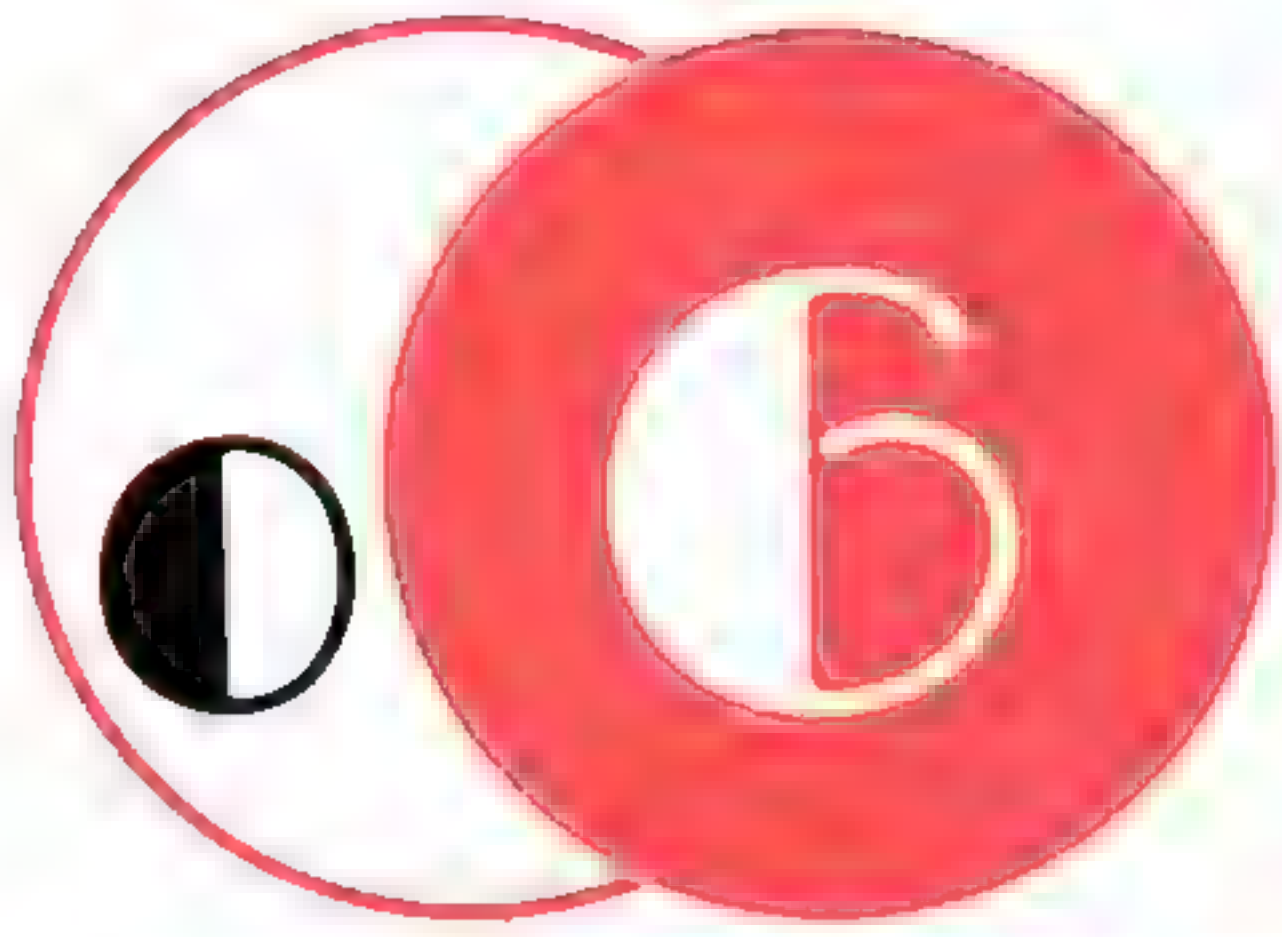
入力装置
- キーボード
- ジョイスティック
- マウス
- ペン

→

本体
- ＣＰＵ
  - 主記憶装置
  - 制御装置
  - 演算装置

→

出力装置
- テレビ
- ディスプレイ
- プリンタ
- プロッタ

補助記憶装置 ↑
- フロッピーディスク
- カセットテープ

パソコンは0か1だけの世界

電流があるかないかだけで判断

パソコンと長くつき合って行こうとするなら、少しむずかしいかもしれないけれど、2進法っていう数字の数え方も知っておいたほうがいい。

ふつうさんすうなどで使っている数え方は、9の次に桁上がりして10になるから、10進法だ。それくらいは知っているよね。

ところが、2進法は0と1の2種類の数字しか使えないから、0、1ときて、次はすぐ桁上がりして10となるのだ。

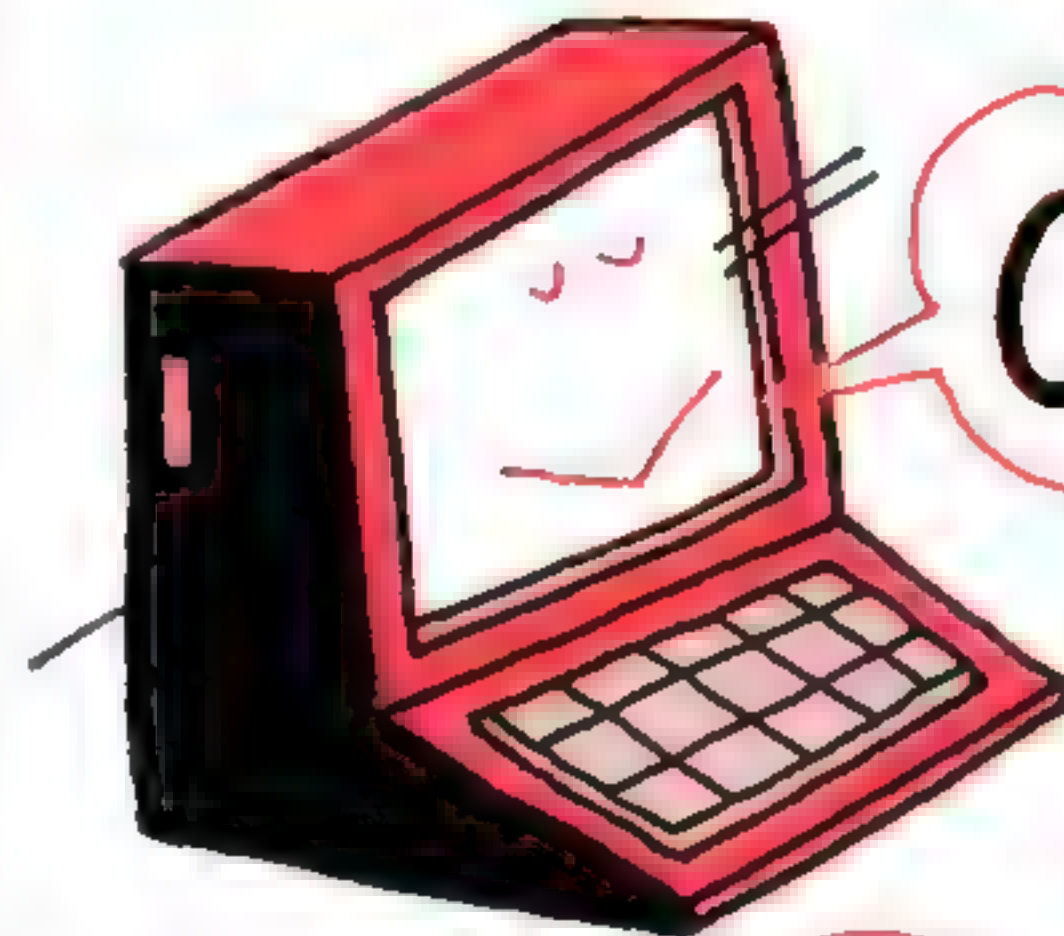
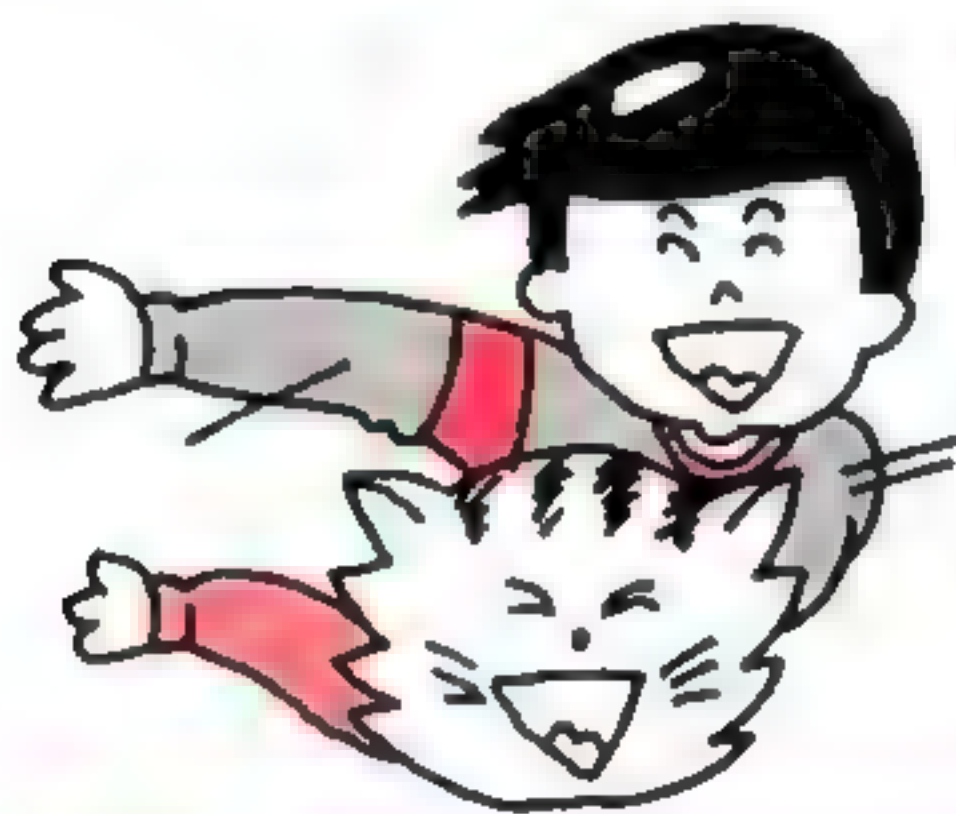
なぜ、そんなめんどうな数え方をしなければならないかって？　それは、コンピュータの世界が、0と1しかない世界だからだ。

パソコンが情報を判断する材料は、電気回路に電流がきていないか、きているかだけだ。それで電流がきていなければ0、きていれば1と表現する。すべての情報を0と1の組み合わせで表現するから、どうしても2進法でなければならないというわけ。パソコンは、電流があるか、ないかだけで判断するというところが、たいせつなのだ。

時間の数え方は
60秒で1分
60分で1時間と
桁上がりするから
60進法なのだな

●10進数と2進数

10進数と2進数の関係

10進数	2進数
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010
100	1100100

パソコンの内部は電流ONとOFFだけの世界だから、ONは1、OFFは0として、2進法だけが動いている。だからといって、それを紙に書いていくときなどは、2進法で書くと桁数ばかり多くなってめんどうだ、というわけで、16進法で書かれることがある。

たとえば、10進数の15は2進数では1111と4桁だが、16進数ではただ1字のFですむからだ。

ビットとバイト

ビットは格納する情報の桁数だ

8ビットパソコンとか、メモリ容量1000Kバイトとかいうように、「ビット」と「バイト」はよくきくことばだが、いったいなんだろう？

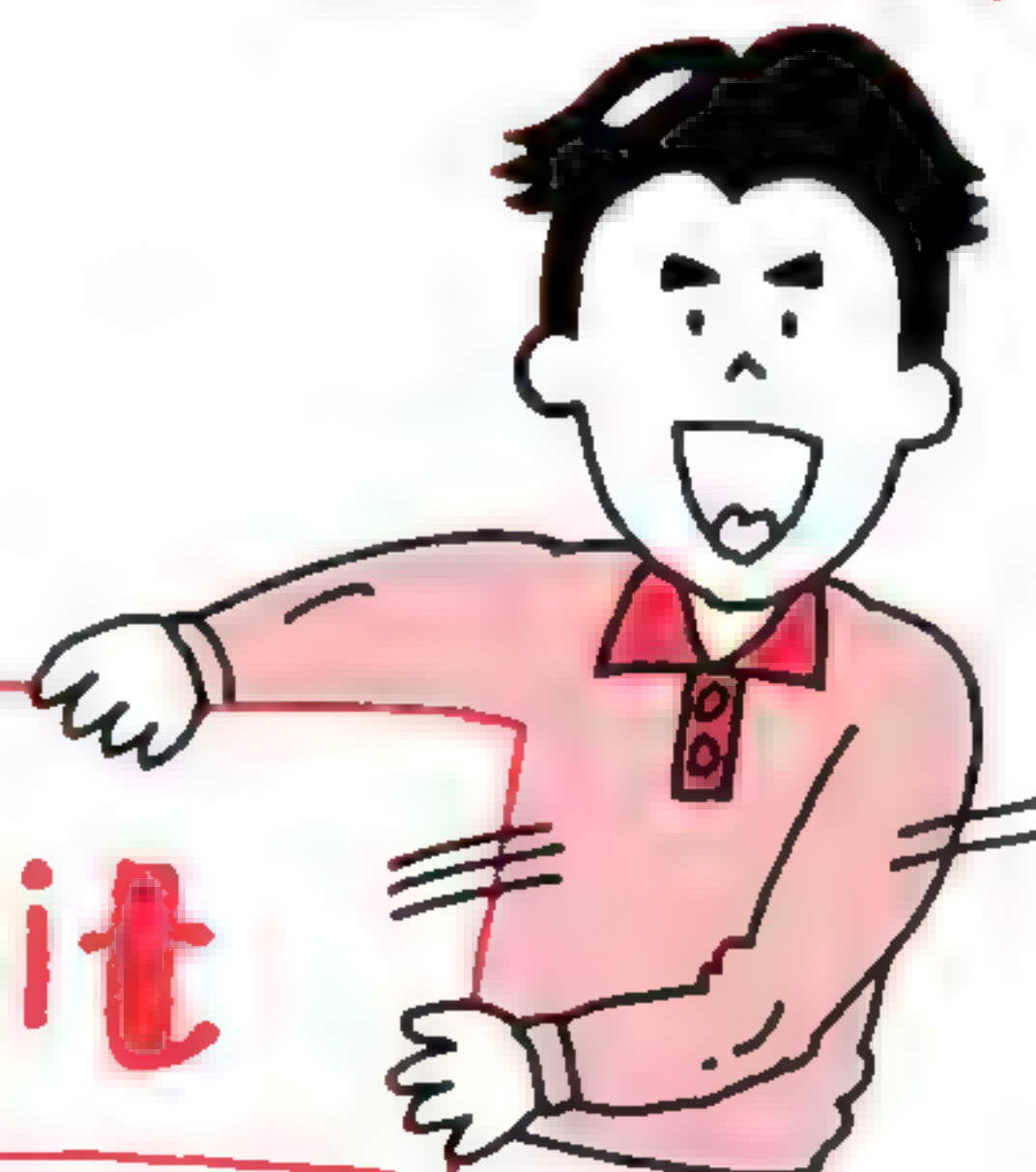
パソコンの情報が、すべて2進数の0と1でできているのは、もう前に書いたから知っているね。ビットは、この2進数の情報の桁数のことだ。

2進数の情報が入っている容量をメモリ（記憶装置）というのだが、そこに格納する容量の単位がビットだと考えていい。

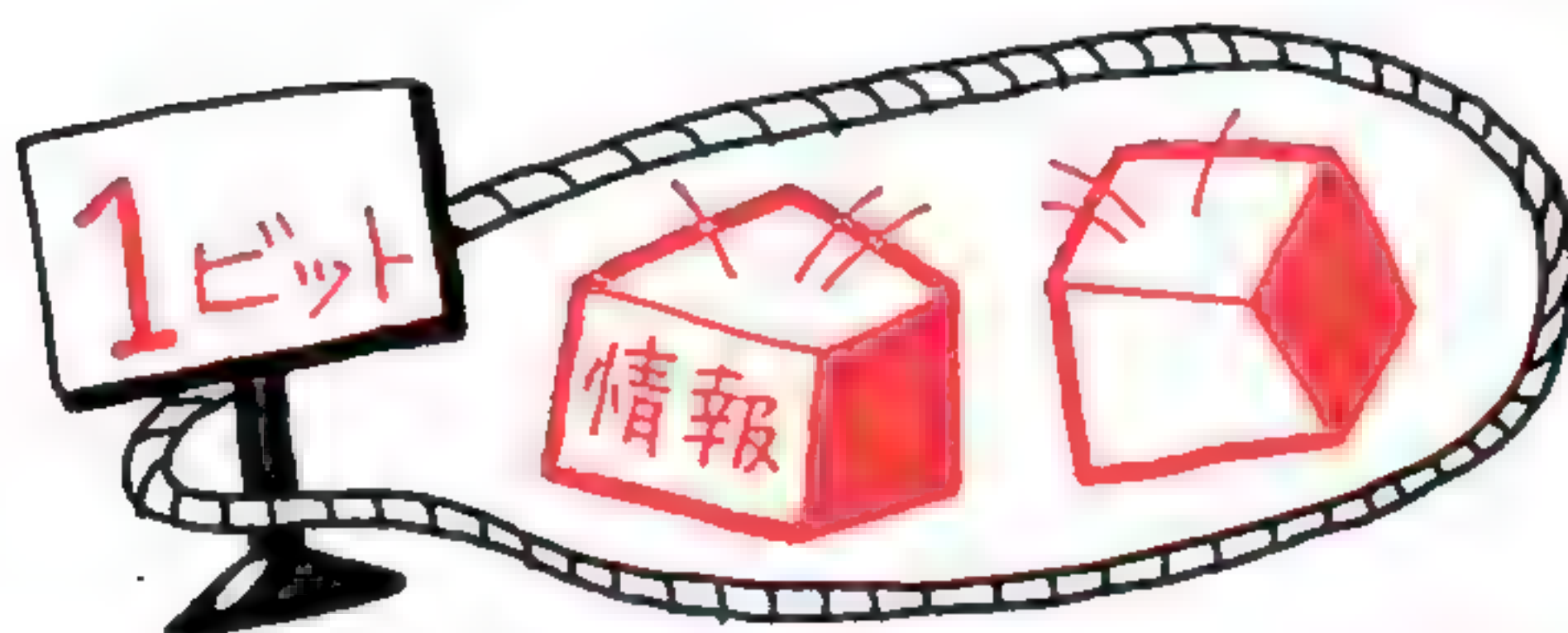
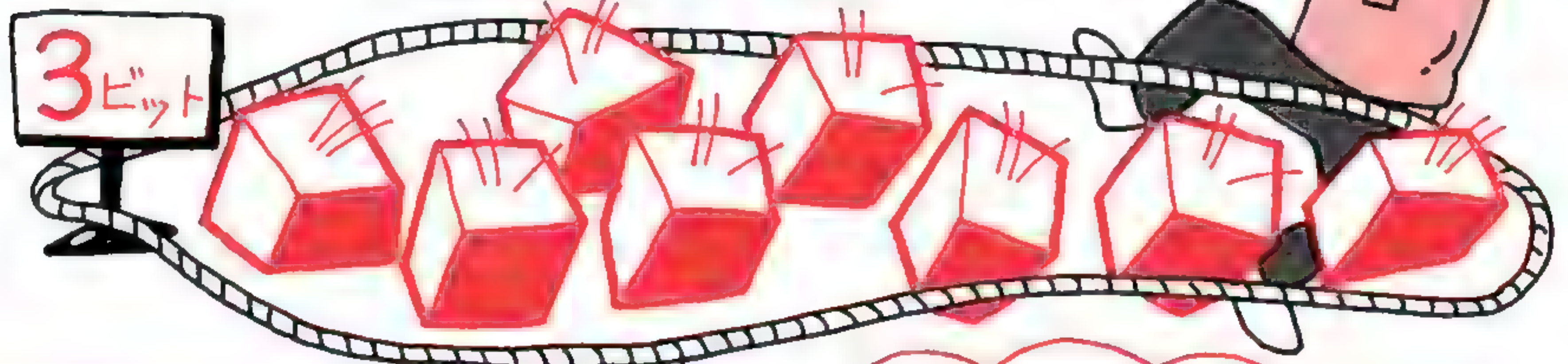
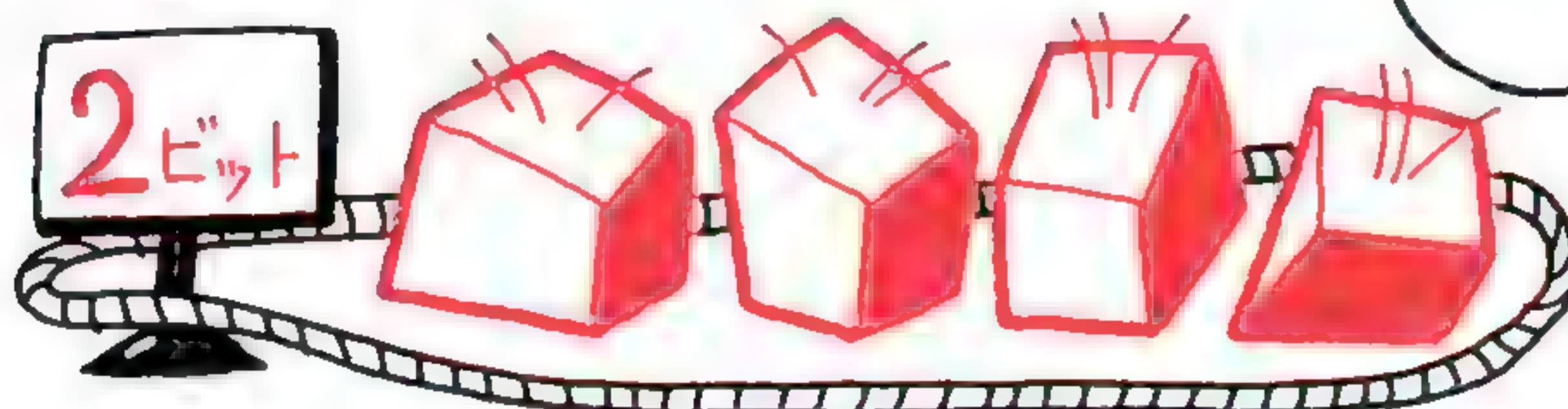
メモリの大きさの単位がビットだが、この8ビットを新しい単位として1バイトという。なぜだろう？　メモリの容量などをビットで数えると数が大きくなってしまうのと、8ビットCPUとか16ビットパソコンなどというように、8ビットを単位として考えると、なにかとべんりだからなのだ。

そのほか、情報量の単位としては、Kバイト、Mバイト、Gバイトもあるよ。

01なら2ビット
0001なら
4ビットなのだ

ビット（Bit）とは
2進数（Binary digit）
の略で桁数のことだ

●各ビットの情報量

各ビットの情報

1ビットの情報	0，1
2ビットの情報	00，01，10，11
3ビットの情報	000,001,010,011,100,101,110,111
4ビットの情報	0000，0001，0010，0011，0100，0101，0110，0111，1000，1001，1010，1011，1100，1101，1110，1111

1ビット増加するごとに
情報量が2倍になる
バイバイゲームだ

情報の大きさを示す単位

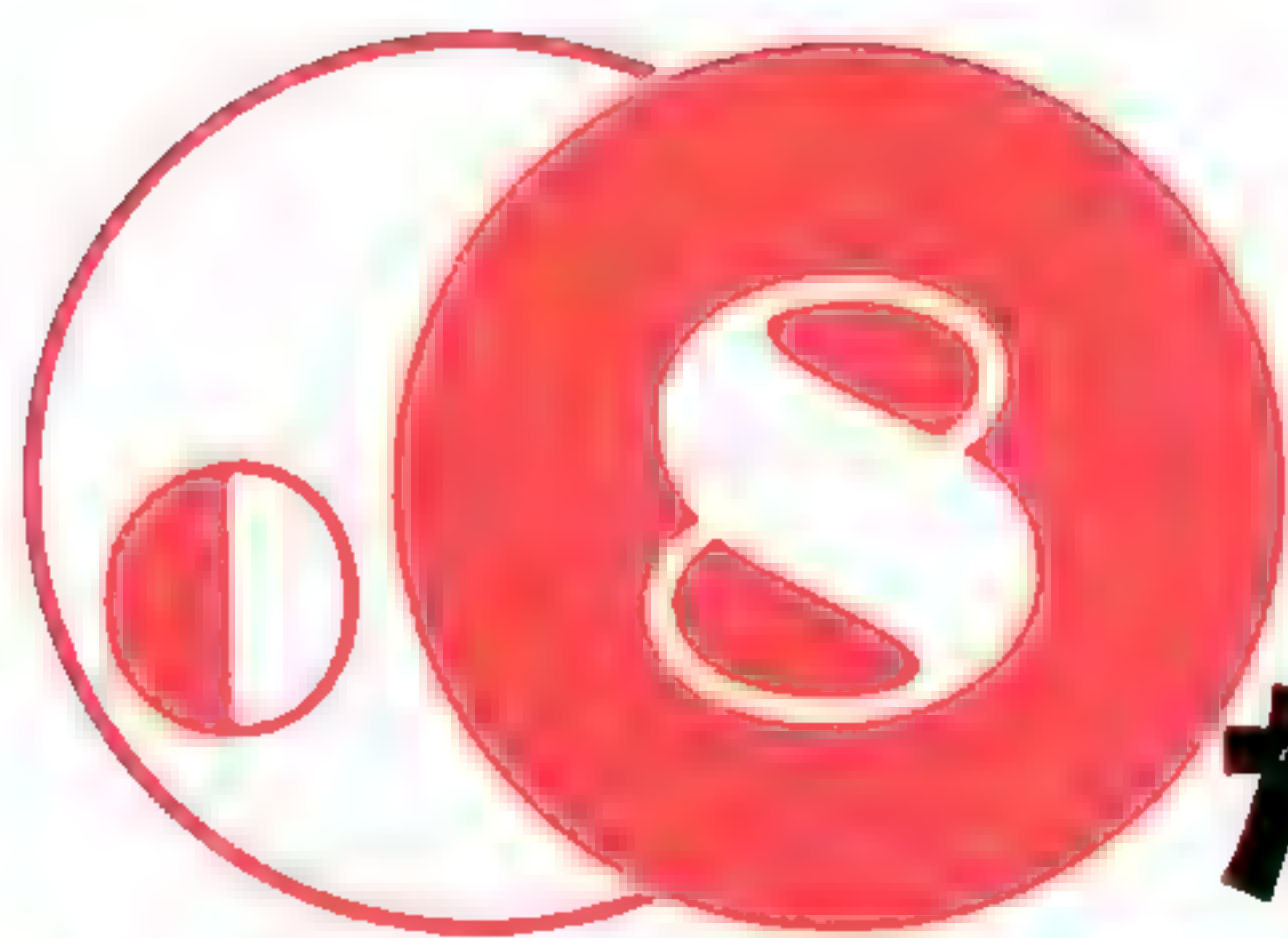
1ビット＝2進数1桁の情報量

1バイト＝8ビット

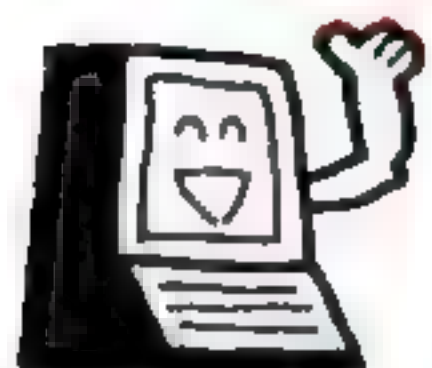
1Kバイト＝1024バイト

1Mバイト＝1024Kバイト

1Gバイト＝1024Mバイト

1ビットの情報は
2進数1桁だから
1か0の2通り
2ビットなら
00、01、10、11の
4通りあるよ

8 ぼくのパソコンだ たいせつに扱ってね

設置する場所の環境に気を配って

操作の前にちょっと待て
扱い方の心得を
よくおぼえてからＧＯ！

念願がかなって、ようやく買ってもらったパソコンだ。たいせつに扱ってね。あまり乱暴に扱うと、パソコンは、きみのいうことをきかなくなってしまうよ。

キーボードのキーは、多少強くたたいても、続けて打ち込んでも、こわれる心配はないから、どんどんキータッチの練習をしてもいいけれど、外気などの環境がわるいと、パソコンはちょっと弱いところがあるから、気をつけてね。

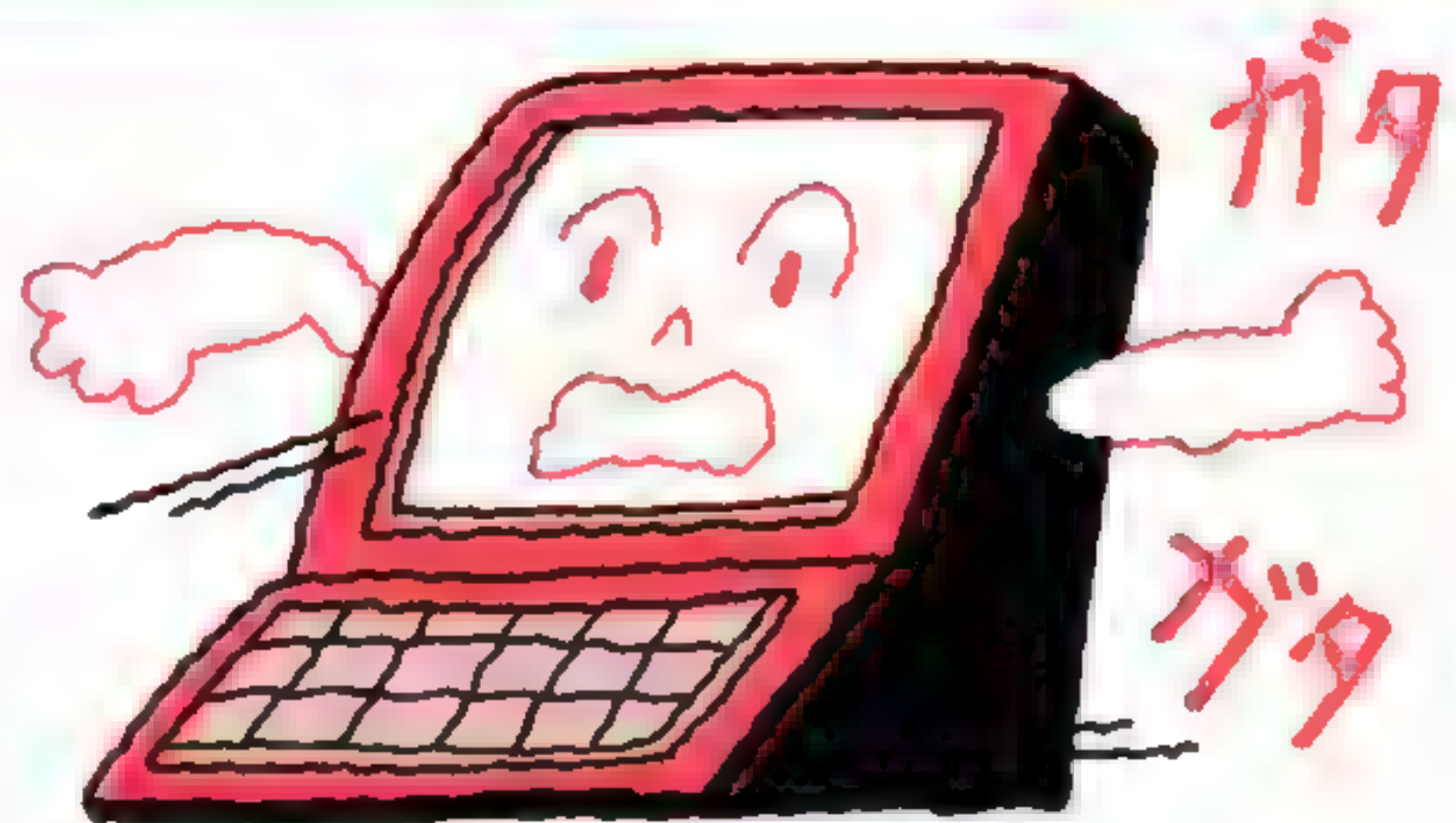
だからといって、パソコンは、ほかの精密機械にくらべて、もろいのかというと、決してそんなことはない。むしろ、ほかの機械よりも構造的には強くできているから、あまり神経質になることはない。

でもやはり、精密機械なのだから、注意するにこしたことはないだろう。これから長くつき合っていくのだから、たいせつに扱って、かわいがってやろう。とくに、振動や熱、ほこり、たばこの煙などはパソコンの大敵なのだ。

でもビビッたりしない
キーボードからは
どんどん打ち込もう

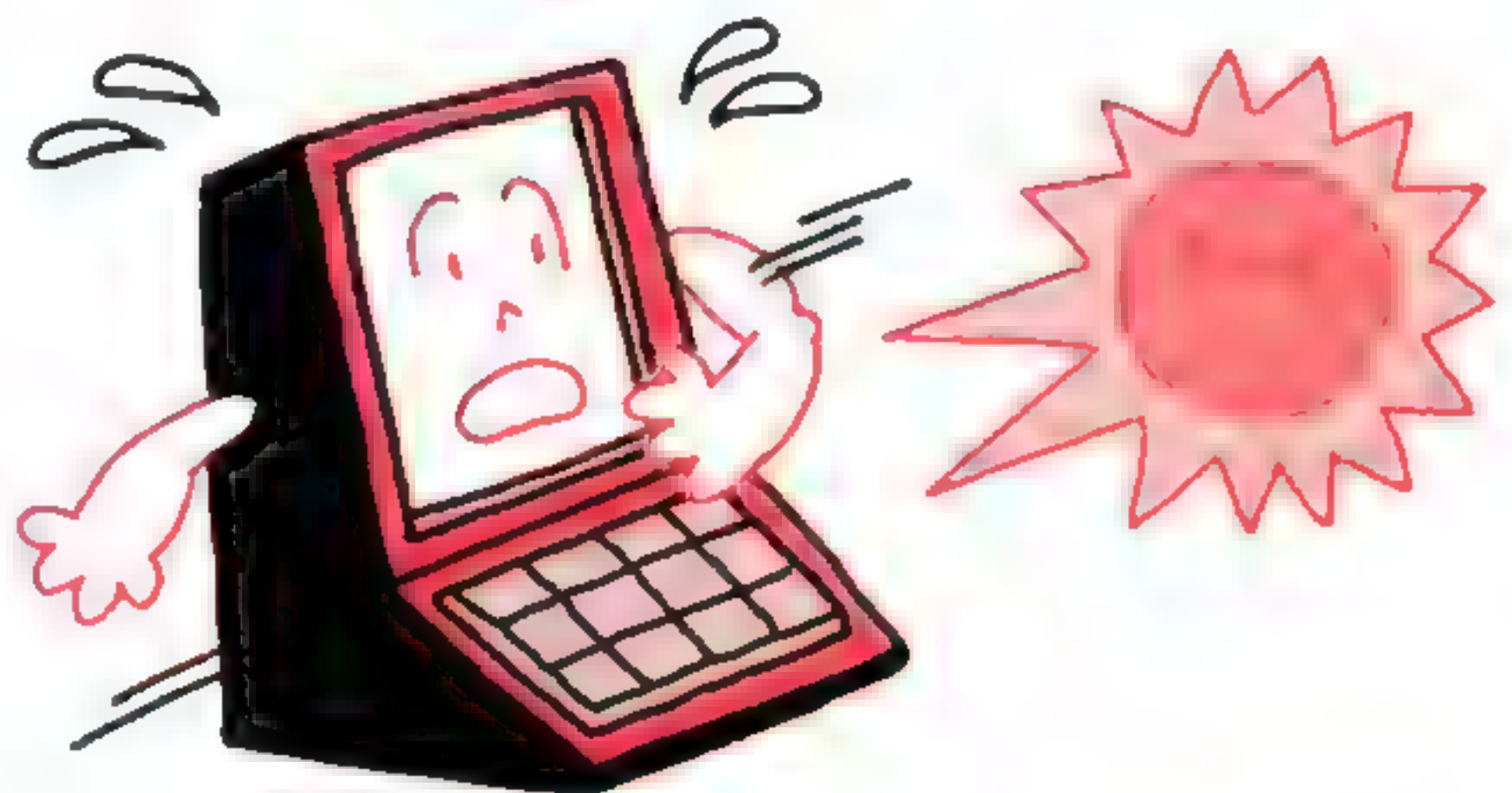
●パソコンの大敵は？

1．常時振動する場所に置かない

精密な電子部品がプリント基板にパターン配線してあるから、振動には弱いところがある。常時振動する場所には、置かないことだ。

2．衝撃を与えない

ぶつけたり、落っことしたりは大敵だ。持ち運びのときは十分気をつけよう。

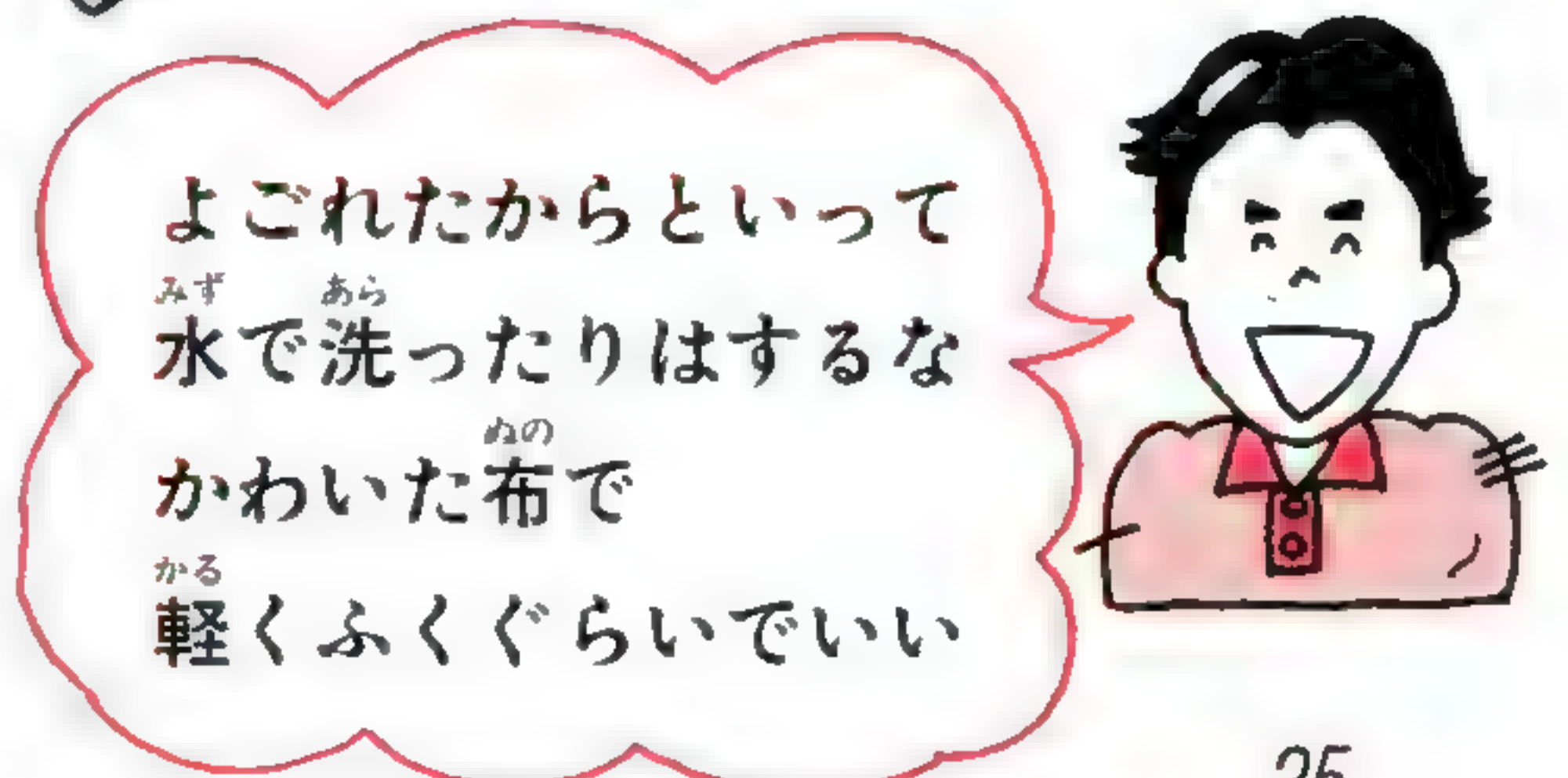
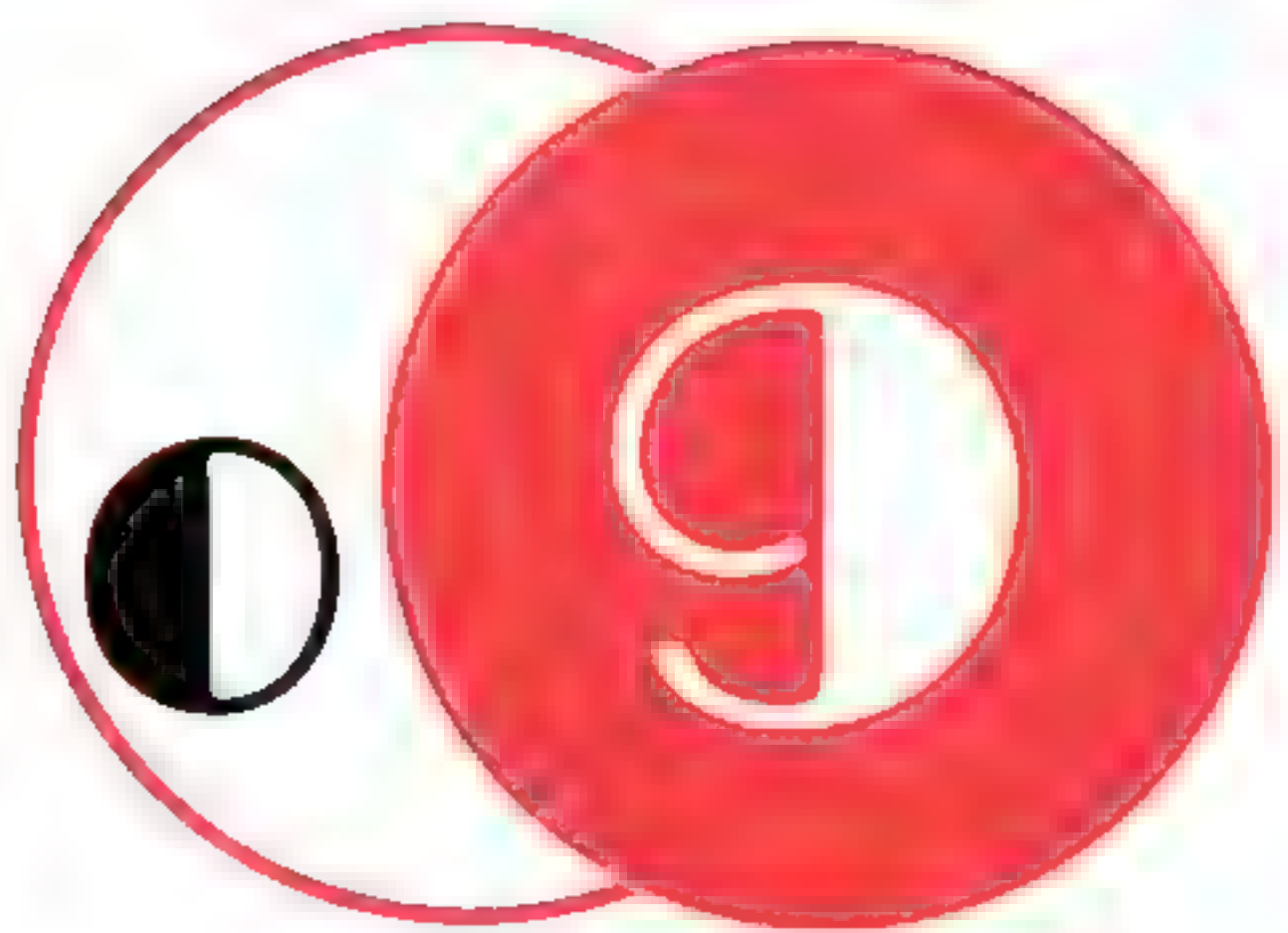
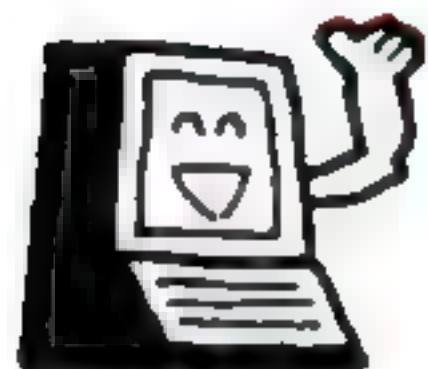
3．直射日光にさらさない

パソコンは急激な温度変化に弱い。とくに、内部温度が急上昇しないようにする。直射日光にさらすなどは、もってのほかだ。

4．煙やほこりを避ける

操作中にたばこを吸ったり、カバーをかけずに部屋をそうじしたりしない。煙やほこりが、すき間から忍び込んで、故障の原因になってしまう。

5．電子機器の雑音に気をつける

電子機器から出る電磁波（ノイズ）の影響で、画面がぶれたり、色調が乱れたりするから注意。

よごれたからといって
水で洗ったりはするな
かわいた布で
軽くふくぐらいでいい

正しく接続しよう

各機種ごとに説明書をよく読んで

パソコンを使用する前に、設置する場所を決めよう。設置したあとで、しょっちゅう移動させなくてもいいように、場所選びを慎重にしよう。

場所が決まったら、本体・キーボードと周辺機器のセッティングだ。

本体・キーボードとテレビ（ディスプレイ）、プリンタ、カセットレコーダ、ジョイスティックなどの周辺機器を接続しなければならない。

接続の方法は、パソコンの各機種ごとに、多少違っている場合があるから、パソコンを買ったときについてくる取扱説明書などをよく読んで、正しく接続することだ。

たとえば、MSXタイプなら、専用ディスプレイのほかに、家庭用テレビとも接続して使用できる。プリンタも、MSX専用タイプ以外のものを接続すると、グラフィックパターンが異なる場合があるなど、十分に知ってから接続しなければならない。次のページで、一般的な接続の方法を説明しておこう。

ディスプレイは
専用テレビか
家庭用テレビかで
接続方法も違うよ

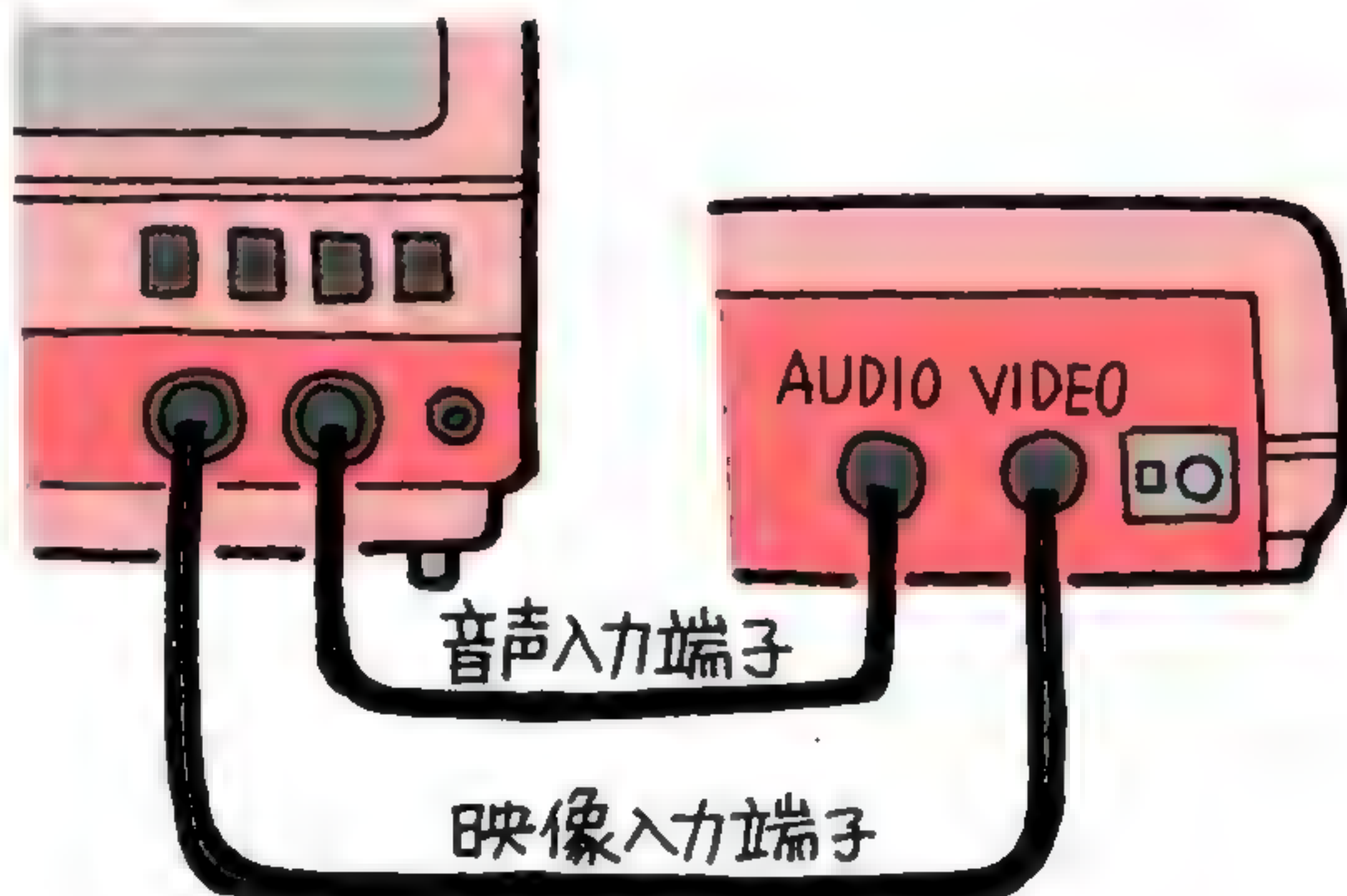
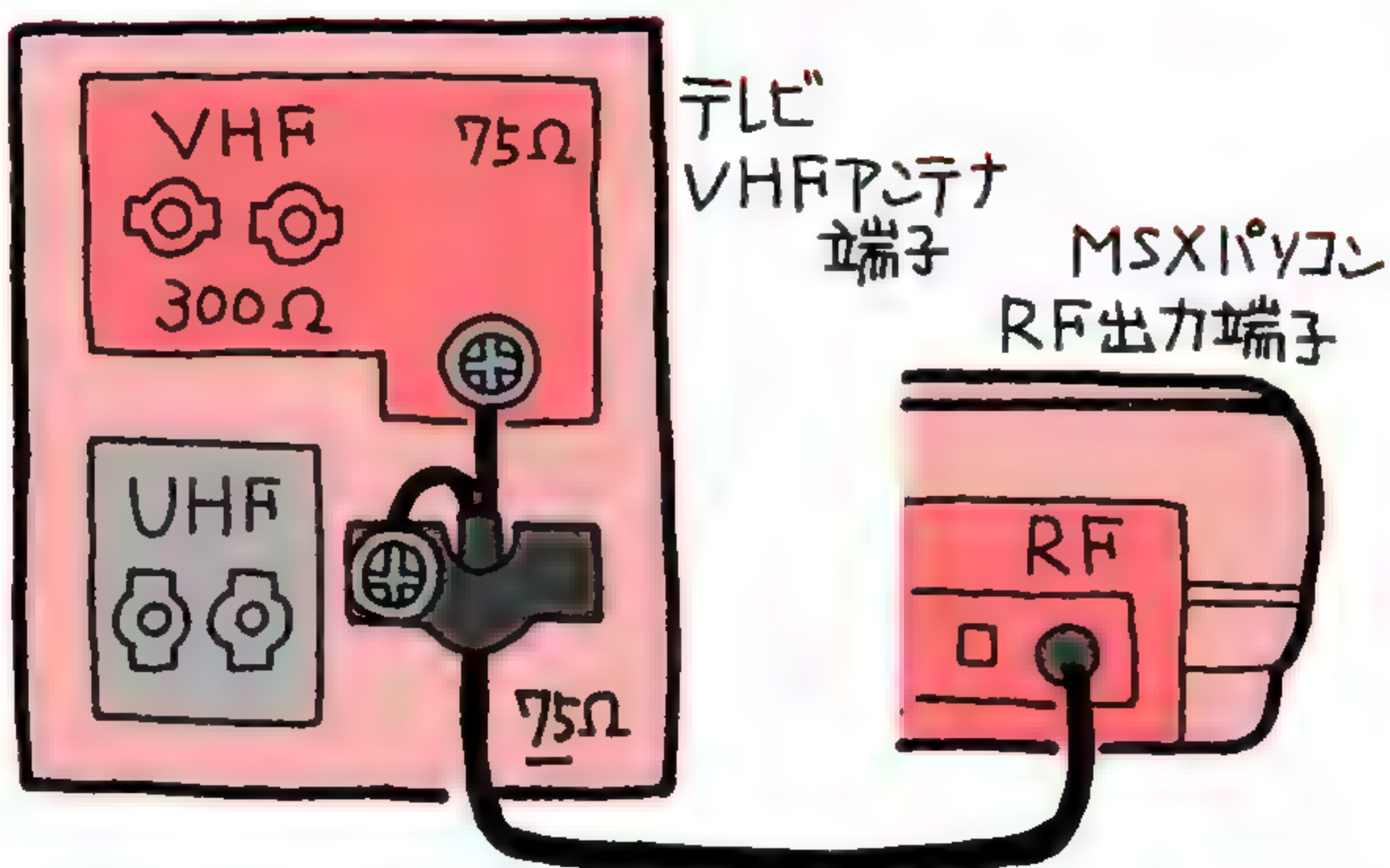
●周辺機器の接続

テレビ（ディスプレイ）の接続

家庭用テレビにも接続できるＭＳＸには３通りの接続方法がある。

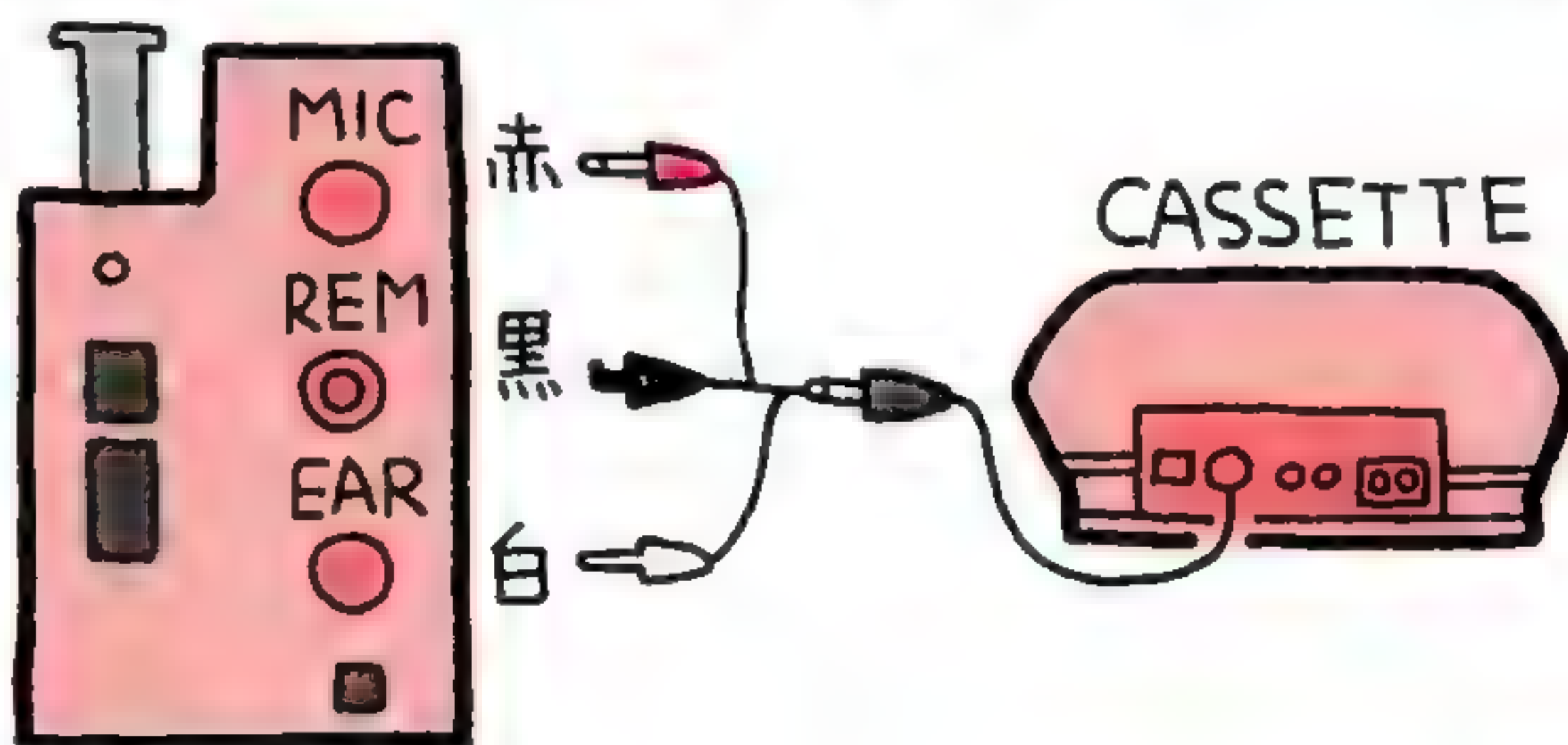
1．家庭用テレビのアンテナ端子に、ＲＦ出力端子の内芯線、入力端子に、外芯線を接続する。

2．テレビの映像入力端子とVIDEO出力端子、テレビの音声入力端子とAUDIO出力端子を接続する。

3．ＲＧＢ端子を持つ機種なら、１と同じ方法で専用テレビと接続する。

ジョイスティック

本体に２個の接続端子があるから、１台だけを使用するなら、１の端子と接続する。

テープレコーダ

本体の入力端子に、OUT（白）、IN（赤）、REMOTE（黒）を接続。

いよいよスイッチONだ

まず周辺機器、最後に本体の順に

本体と周辺機器の接続がすんだら、いよいよ電源スイッチをONして、パソコン操作自由自在だ。といきたいところだが、あわててスイッチONしてはいけないよ。

本体でも周辺機器でも、手あたりしだいにスイッチをONしていっては、故障のもとだよ。ものには順序というものがある。スイッチONの順序も、正しく守ってもらわなければならない。

スイッチONの順序は、まず周辺機器、それから本体だ。テレビ（ディスプレイ）だけが接続されているのなら、まずテレビのスイッチを入れてから、本体・キーボードのスイッチを入れる。

プリンタやテープレコーダなど、ほかの周辺機器が接続されているときは、周辺機器のどれを最初にスイッチONしてもいいけれど、すべての周辺機器をすましてから、本体のスイッチを入れる。

スイッチOFFの順序はその逆だ。まず最初に本体・キーボードのスイッチを切る。それから周辺機器のスイッチを切るという順序だ。

はじめての画面表示は英語だ

本体・キーボードとテレビ（ディスプレイ）だけを接続したとして、実際に電源スイッチを入れてみよう。まずテレビの電源をONして、それから、本体のスイッチONの順序だよ。ここでは、SONYのHIT BITを参考にしよう。

あっ！とおどろくほどのことではないが、画面になにかが表示されたはず。文字だ。しかも、英語だよ。機種が違えば、多少は表示の内容も違うかもしれないけど、まあ、似たりよったりだから、多少の違いには目をつぶろう。

最初から
英語が表示されたって
こわがらなくてもいい
やさしい
英語なのだから

MSX　system
version　1.0

Copyright　1983　by　Microsoft

最初の画面。システムソフトの種類を示している

やがて、最初の画面から、次の画面に変わる。

カーソルの動かし方は
次のページで
よくおぼえてね

SONY　HIT　BIT　system

MENU

●ジュウショロク
●スケジュール
●メモ
●BASIC

カーソルキー でえらんでください RETURN

プログラムのメニューを選択する画面

これで操作はいつでもOKだ

前の画面に表示された文字を、よく見てみよう。MENUと書いてあって、その下に、

- ジュウショロク
- スケジュール
- メモ
- BASIC

と書いてある。そして、その下を見てほしい。

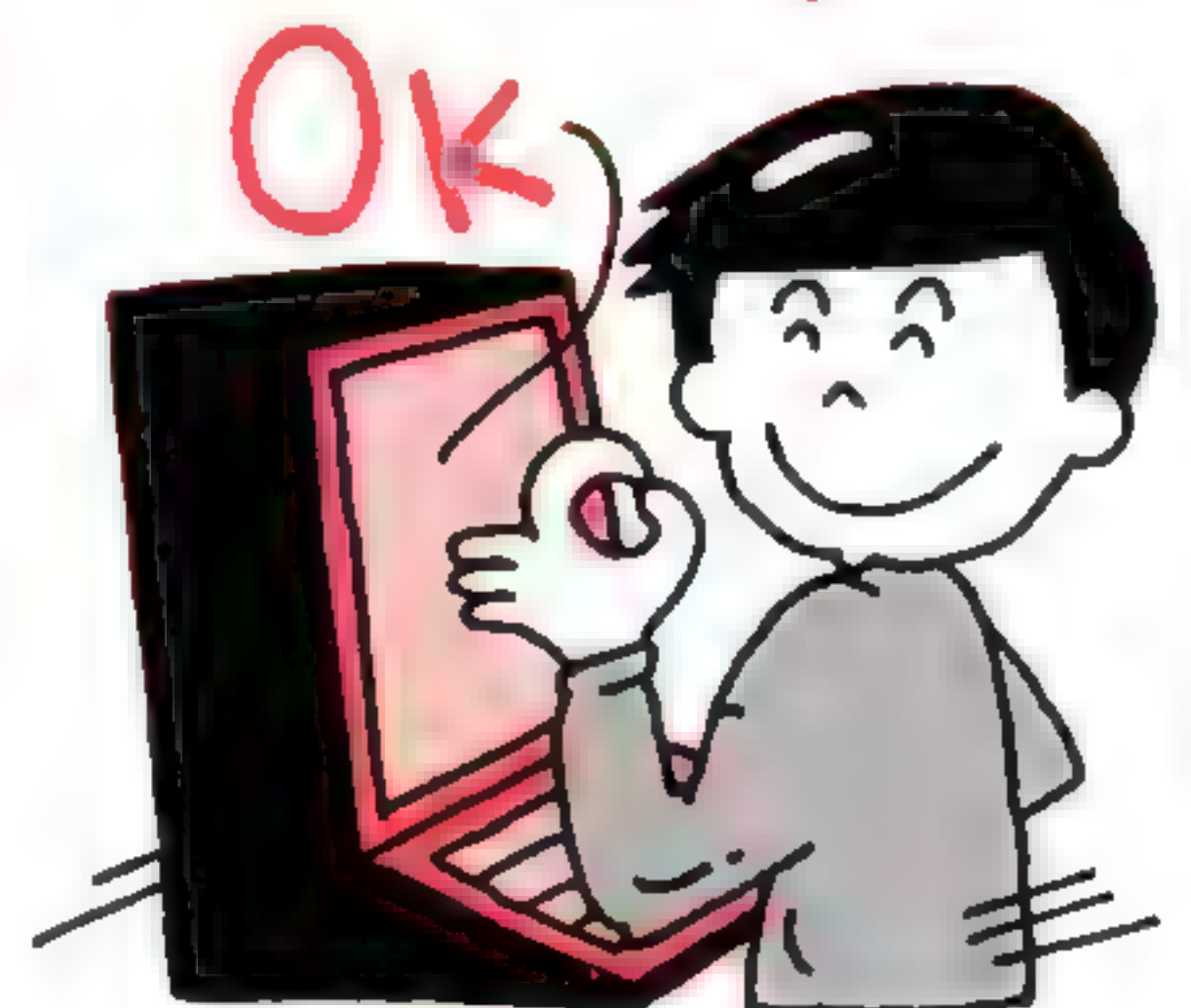
カーソルキー でえらんでください RETURN

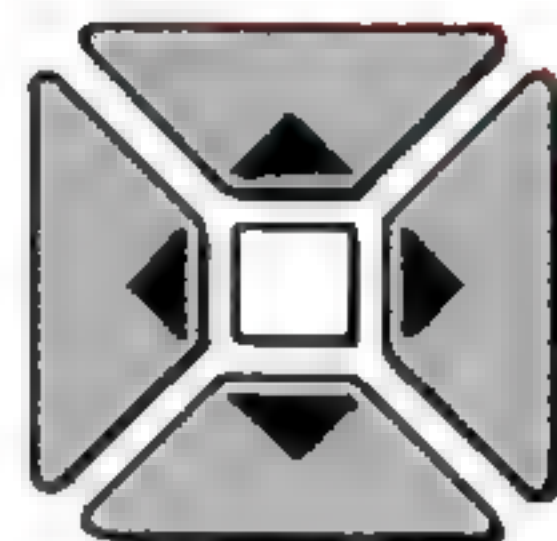
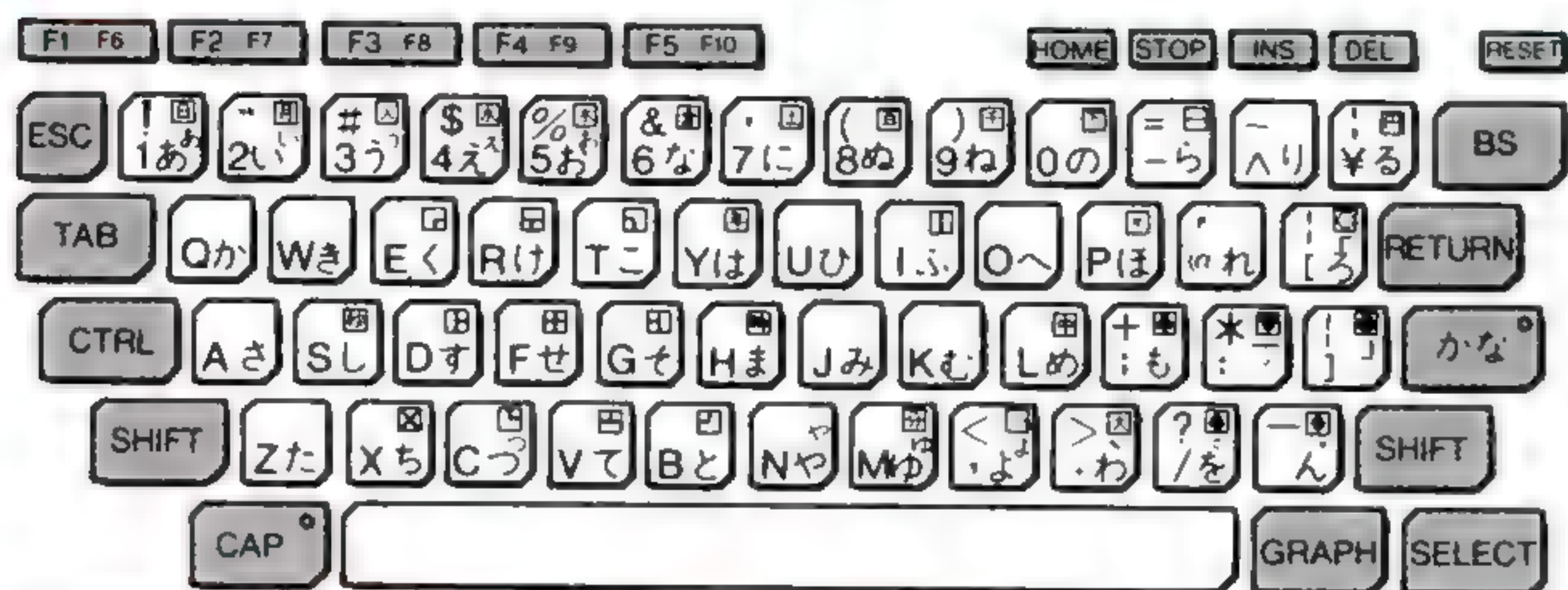
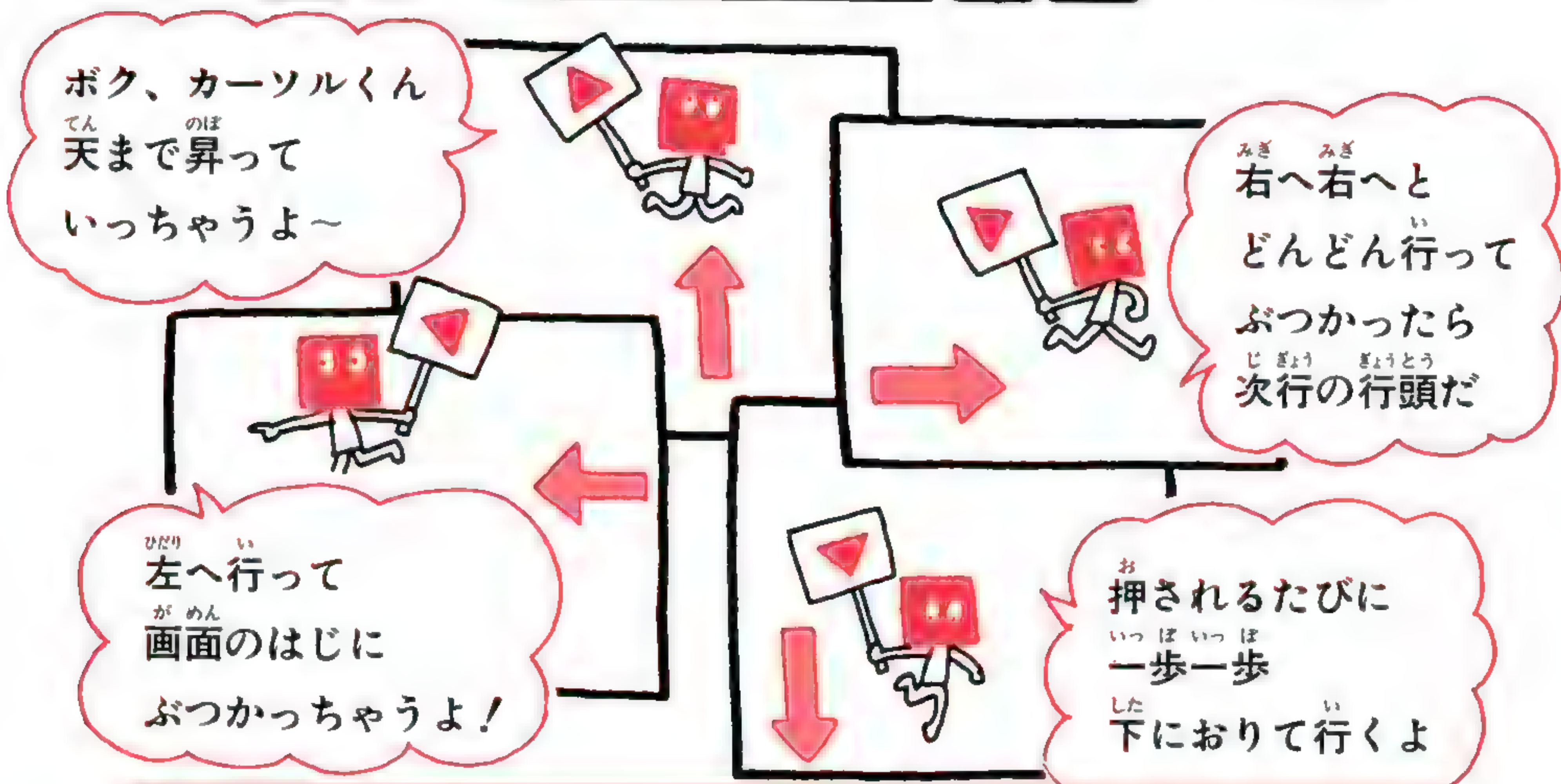
となっているから、そのとおりにしてみよう。カーソルキー▼を押して、画面にあるカーソルマーク（■）を、BASICと書かれた位置に移動させて、RETURNキーを押せばいい。

```
MSX  BASIC  version  1.0
Copyright  1983  by  Microsoft
12431  Bytes  free
OK
■



color  auto  goto  list  run
```

これで、パソコンを操作する準備は、すべて完了したというわけ。あとは、キーボードから、どんどん入力してみよう。

カーソルキーの使い方は次のページにくわしくのっているよ

OKは
スタンバイOKだ
パソコン操作をしてもいいよというわけ

●カーソルキーの使い方

●キーボードとカーソルキー(SONY・HIT BITの場合)

カーソルとカーソルキー

画面にあるカーソル（■）の位置に、キーボードから入力した文字や記号が表示される。逆のいい方をすれば、入力する文字や記号が表示される位置を示すのが、カーソルだ。

カーソルは、画面の中で自由に動かすことができる。カーソルを動かすのが、カーソルキーだ。カーソルキーは矢印のついた▲▼▶◀の4個のキーだ。これらのキーを押すと、カーソルは矢印の方向に移動する。

キーボードから文字や記号の入力

画面をきれいにしたあとで

画面にＯＫと出たら、操作準備ＯＫだから、どんどんキー入力してもいいのだが、画面がよごれたままでは気に入らない。画面のそうじをして、きれいな画面にしてから、レッツゴー！だ。画面にＯＫと出たら、

CLSとキーを押して

RETURNキーを押す

画面がきれいになって、表示されているのは、左上にＯＫとカーソル（■）だけとなったはずだ。

SHIFTキーを押しながらHOMEキーを押しても画面はきれいになるよ

さあ、いよいよ、文字や記号のキーを押して、文字や記号を表示させてみよう。でも、ただやみくもにキーを押しても、思った文字や記号は表示されないよ。アルファベットや数字、かたかな、ひらがな、記号にマークなど、文字や記号の種類ごとに、キーの押し方が違うのだ。

たとえば[あ]のキーを見てみよう。

ひとつのキーにいろいろ書いてある。これらから、入力したり、文字を選ばなければならないのだ。

●文字や記号の入力方法

文字や記号は、次のようにキーを打って入力する。

英小文字――→［英字］

英大文字――→［CAP］［英字］

ひらがな――→［かな］［文字］

ひらがな小文字――→［かな］［SHIFT］＋［文字］

かたかな――→［かな］［CAP］［文字］

かたかな小文字――→［かな］［CAP］［SHIFT］＋［文字］

数字――→［数字］

記号（キーの左下に表記）――→［記号］

記号（キーの左上に表記）――→［SHIFT］＋［記号］

グラフィック記号→［GRAPH］＋［記号］

●キーに表記された位置での打ち分け

A＝数字、英小文字、記号（－＾￥＠［；：］，．／）――→［キー］

B＝記号（！”＃＄％＆’（）＝～¦’｛＋＊｝＜＞？＿）

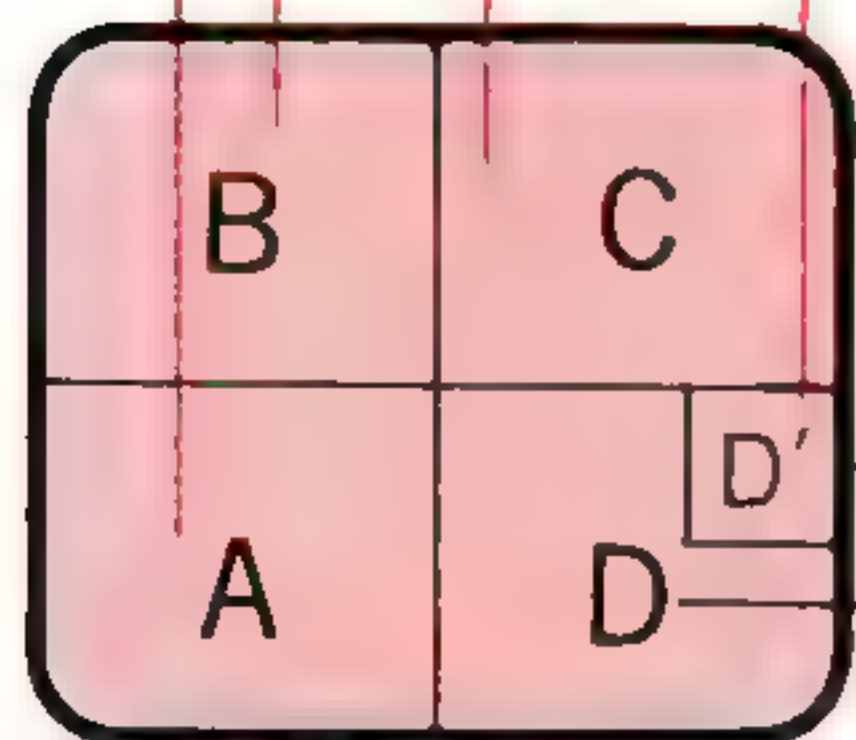
――→［SHIFT］＋［キー］

C＝グラフィック記号――→［GRAPH］［キー］

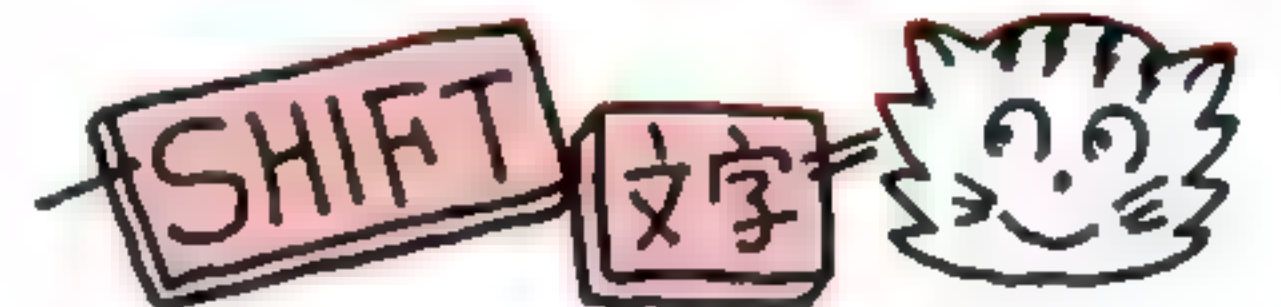
D′＝かたかな小文字、ひらがな小文字、記号（「―」、

。・）――→［かな］［SHIFT］＋［キー］（ひらがな）

または［かな］［CAP］［SHIFT］＋［キー］

（ひらがなの小文字、記号）

B　C
A　D　D′

D＝ひらがな――→［かな］［キー］

かたかな――→［かな］［CAP］［キー］

●ファンクションキーの使い方

ファンクションキーは、あるまとまった命令を入力したと同じ機能を持つキーだ。はじめに電源を入れたとき[F 1]～[F 10]までのファンクションキーは、キーを押すだけで次のような機能を果たすのだ。

[R][U][N][RETURN]
とするよりも
[F 5]キーをワンタッチ
かんたんかんたん

●電源投入時のはたらき

[F 1]──→COLOR
[SHIFT]＋[F 6]──→COLOR　15, 4, 7[RETURN]
[F 2]──→AUTO
[SHIFT]＋[F 7]──→CLOAD"
[F 3]──→GOTO
[SHIFT]＋[F 8]──→CONT[RETURN]
[F 4]──→LIST
[SHIFT]＋[F 9]──→LIST[RETURN]
[F 5]──→RUN[RETURN]
[SHIFT]＋[F 10]──→CLS　RUN[RETURN]

よく出る命令を
定義しておけば
楽チンチン

◆ファンクションキーに新しい機能を定義する

例　[F 1]キーにPRINTを定義する

KEY　1, "PRINT"[RETURN]

定義したい命令語
定義したいファンクションキーの番号

プログラムは保存しておこう

くり返し利用するために

CSAVE&CLOAD●保存したり読み込んだり

せっかく入力したプログラムも、電源を切れば、はかなく霧のように消えてしまうのがパソコンだ。とっても苦心して作ったプログラムなのだから、それを保存して、必要なときに、いつでも使えるようにしなければ、なんのためのパソコンか、といいたくもなるだろう。

これを実現してくれるのが、CSAVEとSAVEだ。

CSAVEは、プログラムをカセットテープに記録して保存する。SAVEは、フロッピーディスクに保存する命令だ。

そして、テープに記録したプログラムは、必要なときにテープから呼び出して、パソコン本体に読み込まなければならない。CLOADがその命令だ。フロッピーディスクに記録したものを呼び出すならLOAD命令だ。

とにかく、苦心して作ったプログラムは、CSAVEかSAVEで、保存しておくことだ。

カセットテープに記録するときのテープはふつうの音楽用テープでいいんだよ

●CSAVE命令とCLOAD命令の使い方

●プログラムができたら、名まえをつけよう

プログラムの名まえは6字以内だ。6字以上の名まえをつけてもいいが、MSXなら、6字までは読み取って、7字めからは、あってもなくても、ないのと同じだ。

たとえば、

マイコンOKコレクション

ここまで登録される（マイコンOK）

というプログラム名なら、「マイコンOK」までが、登録されたプログラム名だ。

1．キー入力　**CSAVE　"マイコンOK"**

2．テープレコーダの録音スタート

3．[RETURN]

●FOUNDとOKが出たらLOAD　OK

テープに保存したプログラムを読み出すなら、

1．電源を切って10秒ほど待って、もう1度電源を入れる。

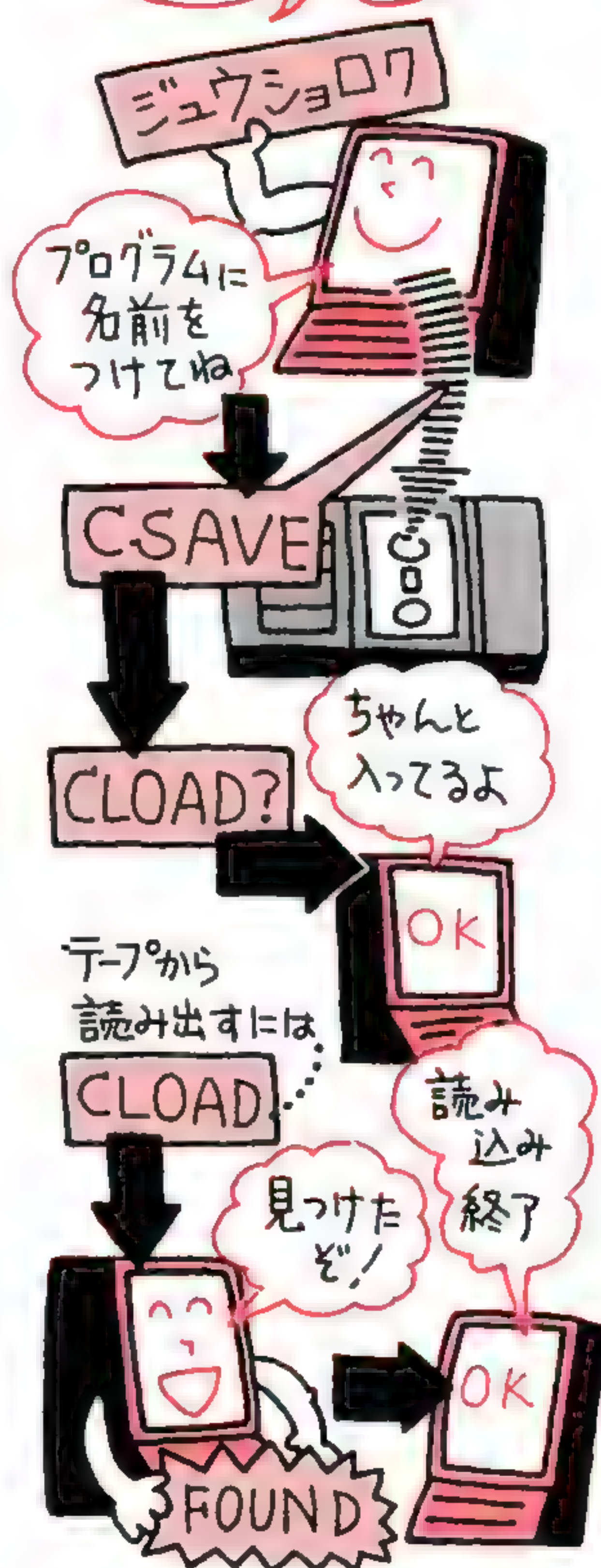
2．キー入力　**CLOAD　"マイコンOK"**（読み込むプログラム名）

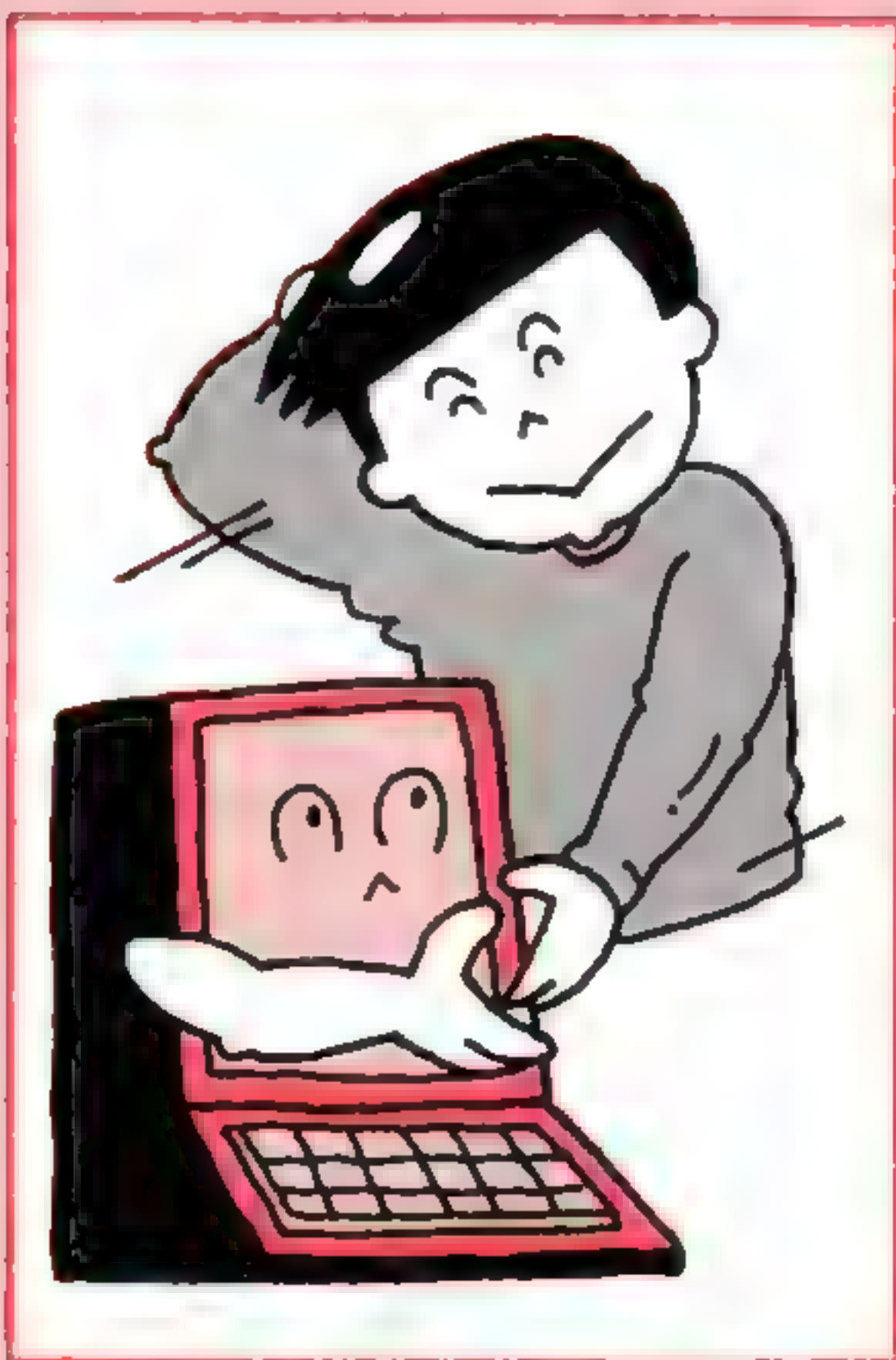
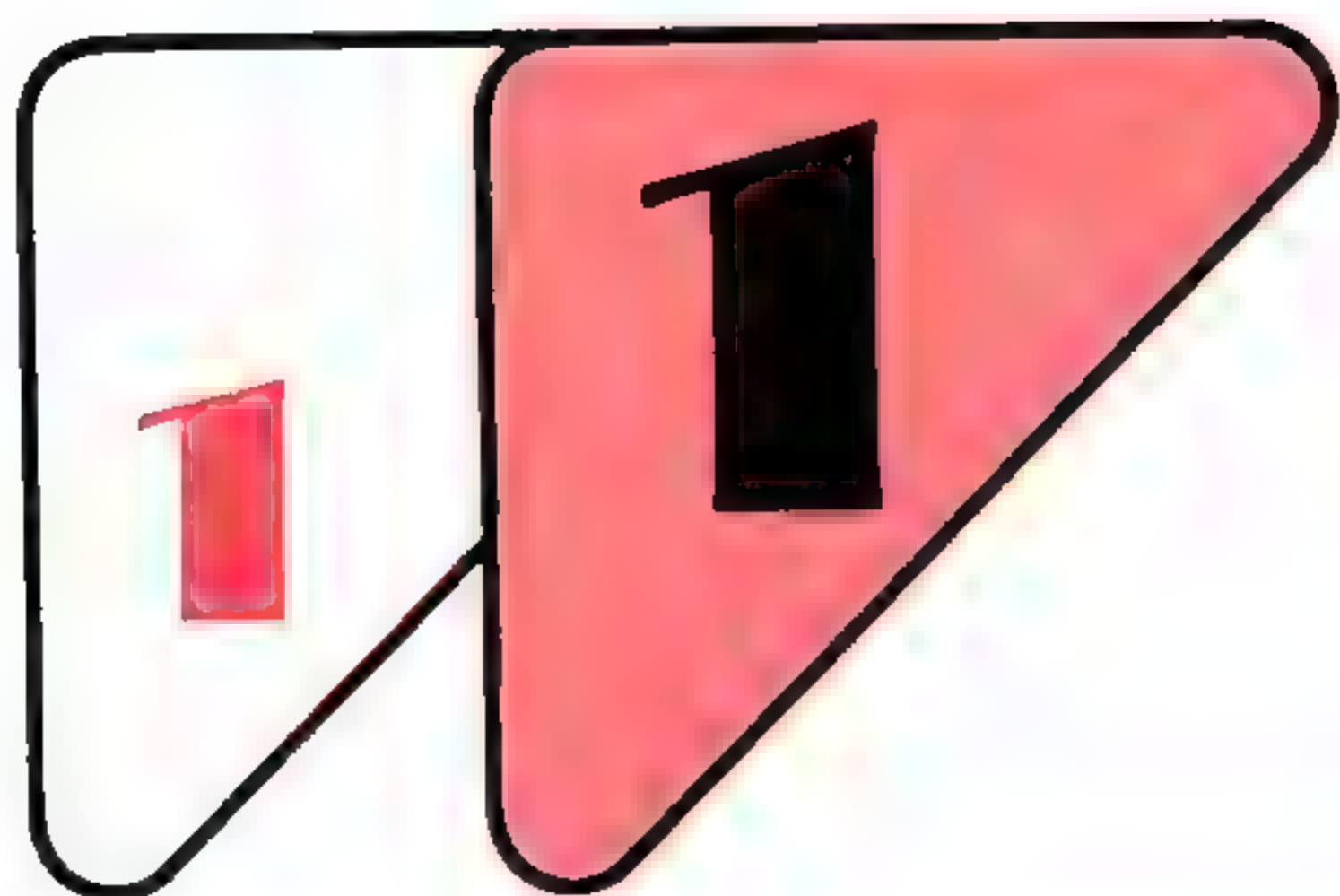
3．[RETURN]

テープの中に、読み込むプログラム名が見つかったとき、FOUNDと表示される。

OKと出たら、プログラムの読み込みが終了したというサイン。

CSAVEが終わったら、電源を切る前に、プログラムがほんとうに入っているかを確認すること。

CLOAD？[RETURN]キーでOKが出ればよし

パート1

パソコン操作への第一歩

パソコン計算機

計算記号をおぼえよう

PRINT命令その1 ●計算して答えを画面に表示する

キーボードのことがだいたいわかったら、とにかくパソコンに何かしごとをさせてみようよ。

コンピュータは、もともと大型計算機として開発されたものだから、計算は得意中の得意。たとえば、地球のまわりをまわっている人工衛星の軌道や打ち上げるロケットの軌道も、むずかしい科学技術の計算も、みんなコンピュータの役目だ。

といっても、ここではそんなむずかしい計算をしようというんじゃない。きみたちの勉強や生活に役立つように、パソコンを計算機として使ってみようということなんだ。

この計算をしてくれるのがPRINT命令だ。

たとえば、3＋5なら、

[P][R][I][N][T]　[3][+][5]

とキー入力して[RETURN]キーを押す。

10－4なら、

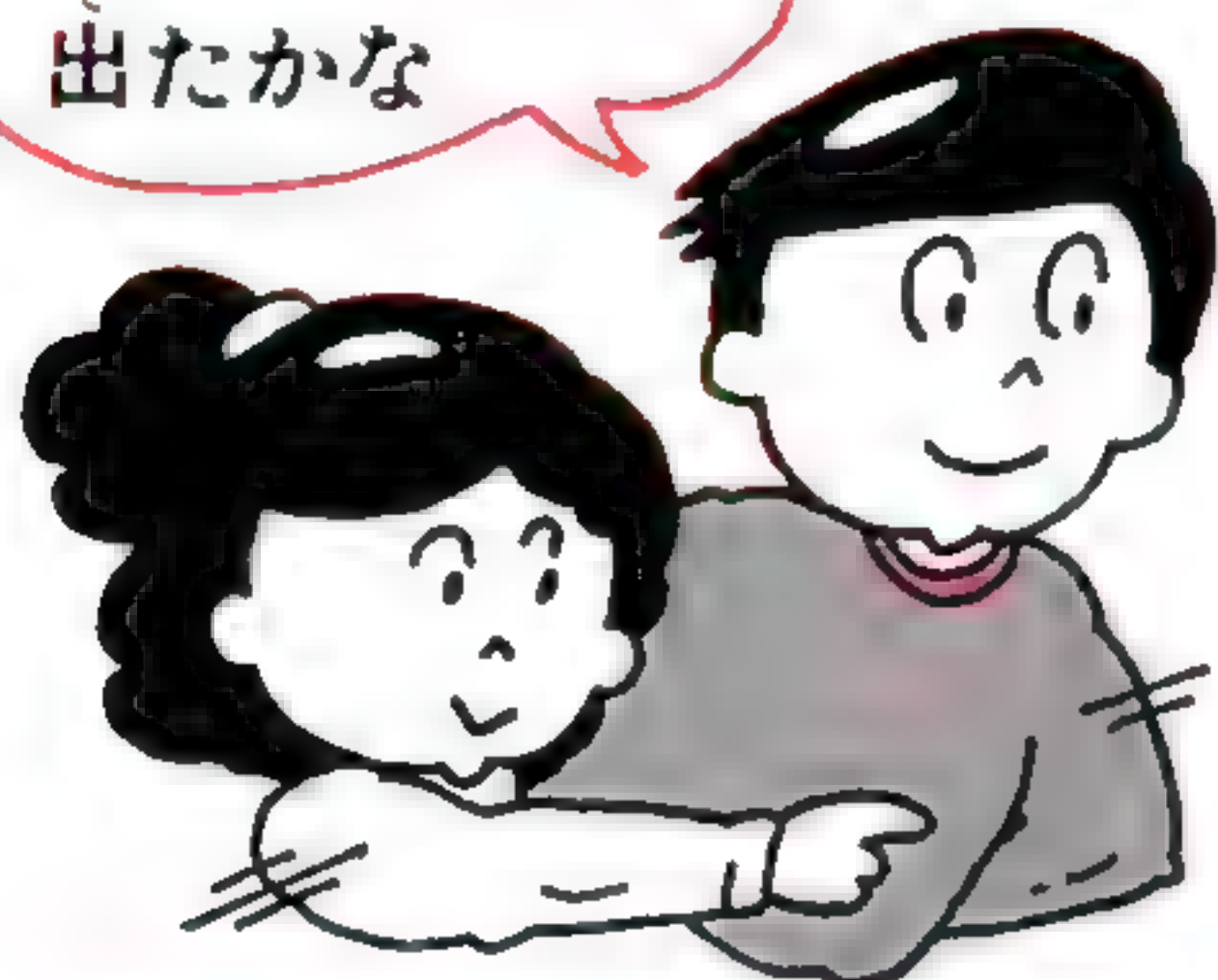
[P][R][I][N][T]　[10][－][4]

とキー入力して、同じく[RETURN]キーを押すんだ。

[RETURN]キーを押し忘れると？

画面には文字が出ているけど、コンピューータに命令が伝わらないよ。注意してね

●計算問題のいろいろ

さあ、画面にはどういうふうに出たかな

```
PRINT  3+5
 8
OK
PRINT  10-4
 6
OK
■
```

1行ごとにRETURNキーを押さないと、計算してくれないよ

RETURNキーを押すのは、「命令を実行しなさい」という意味なんだ

```
PRINT  8/2+5*6
PRINT  (30+5)/7
PRINT  (9*(5-2))/3
PRINT  5^3+4^4
PRINT  9^4/(27*3)
```

もう少しむずかしい計算をやらせてみよう。たし算、ひき算よりかけ算、わり算を先にするのはわかるね

コンピュータの計算記号

＋	……………タス	(例) 5＋3
－	……………ヒク	(例) 10－5
*	……………カケル	(例) 6*2
／	……………ワル	(例) 8／4
^	……………べき乗	(例) 2^3 (2^3)
(　)	……カッコ	(例) 5*(9－2)

(　)カッコがついているときは、内がわのカッコから計算する

直接話したりプログラムで話したり

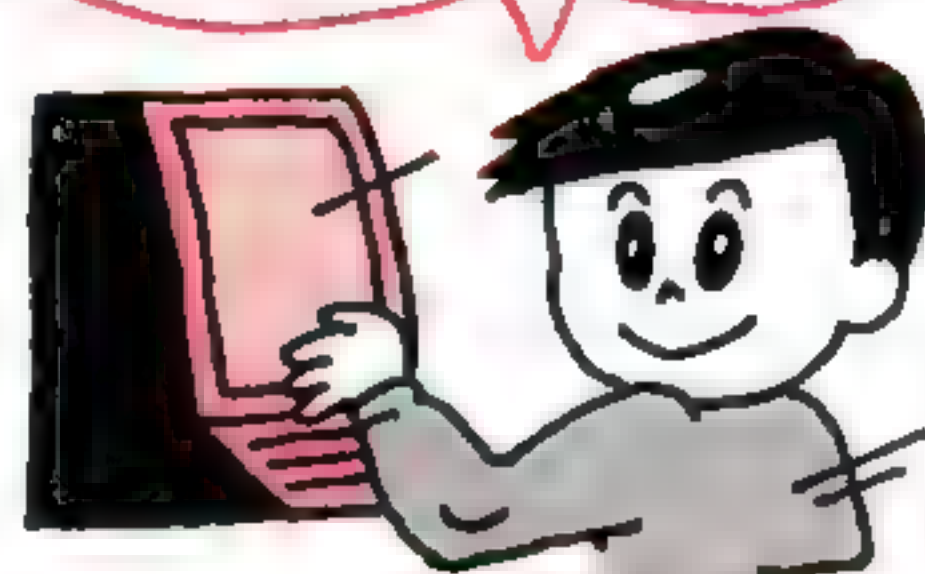
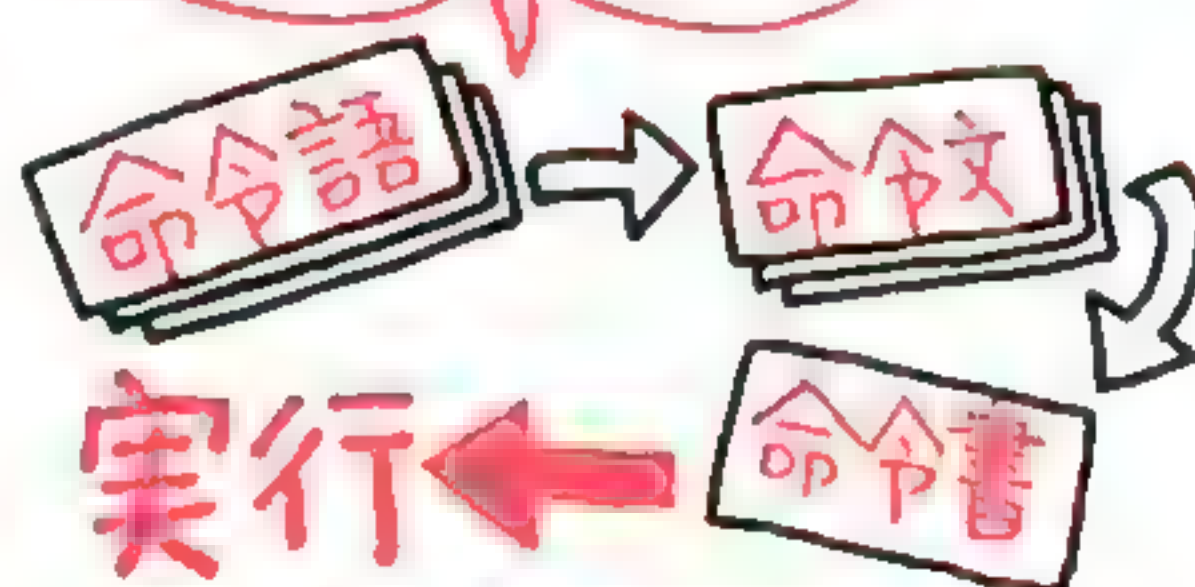
CLS命令●画面をきれいにそうじしてから

0章で、[C][L][S]とキー入力して[RETURN]キーを押すと、画面がきれいにそうじされたね。CLS命令は、それまで画面に表示されていたものをすべて消して、きれいな画面にするための命令なんだ。カーソルは左上に戻るよ。

では、まず画面をきれいにして…………。

キーボードから直接命令を入力して実行させる方法がダイレクトモード

●パソコンに仕事をさせる2つの方法

パソコンに仕事をさせる方法は2通りある。ひとつは、今やった計算のように、直接キーボードから命令を入力して、[RETURN]キーを押し、その場で実行させる方法。もうひとつは、させたいしごとをプログラムにして、まずそれをパソコンに送り込む。そのあと[R][U][N]とキー入力し、[RETURN]キーを押して実行させる方法だ。

プログラムを作ってそれを実行させる方法をプログラムモードというんだ

●プログラムってな〜に？

プログラムは、パソコンにしごとさせるための命令書のようなものだ。この命令書はいくつもの命令文からできていて、しごとの順番にしたがって、1行ごとにきちんと番号がつけられて並んでいる。

まず、命令語を使って命令文を作る。次に、その命令文をどうやって並べて命令書を作るか。これがプログラミングなのだ。

プログラムを実行させる命令が、RUN命令である。

●プログラムを作ってみよう

さっきの計算をプログラムにしてみよう

```
10  CLS
20  PRINT  8/2+5*6
30  PRINT  (30+5)/7
40  PRINT  (9*(5-2))/3
50  PRINT  5^3+4^4
```

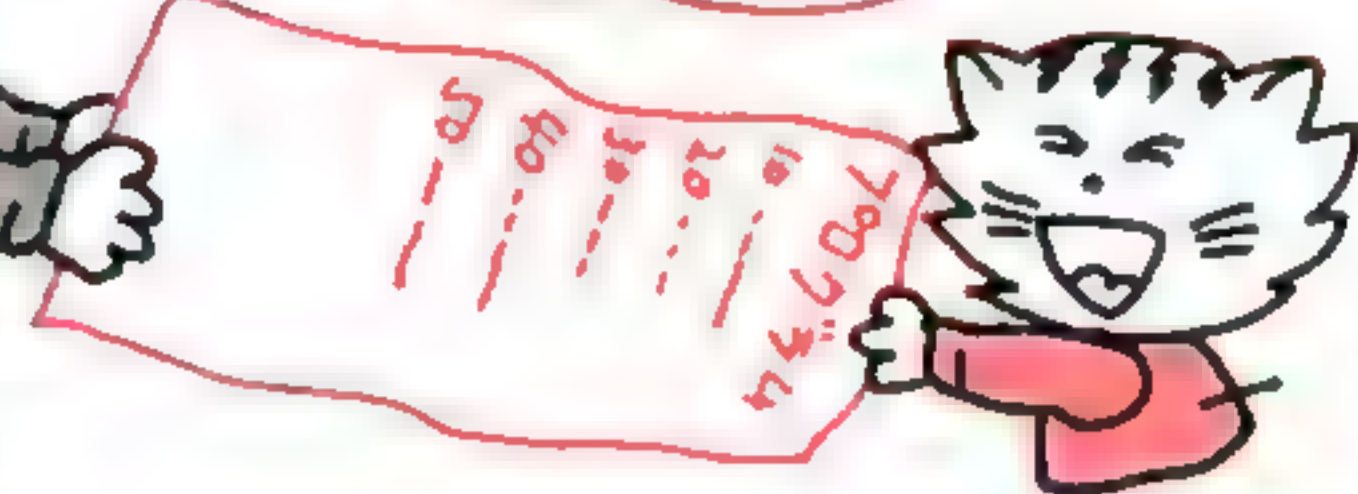
CLS命令を最初に入れておくと、プログラムの実行のときに、自動的に画面をそうじしてくれるよ

```
34
5
9
381
```

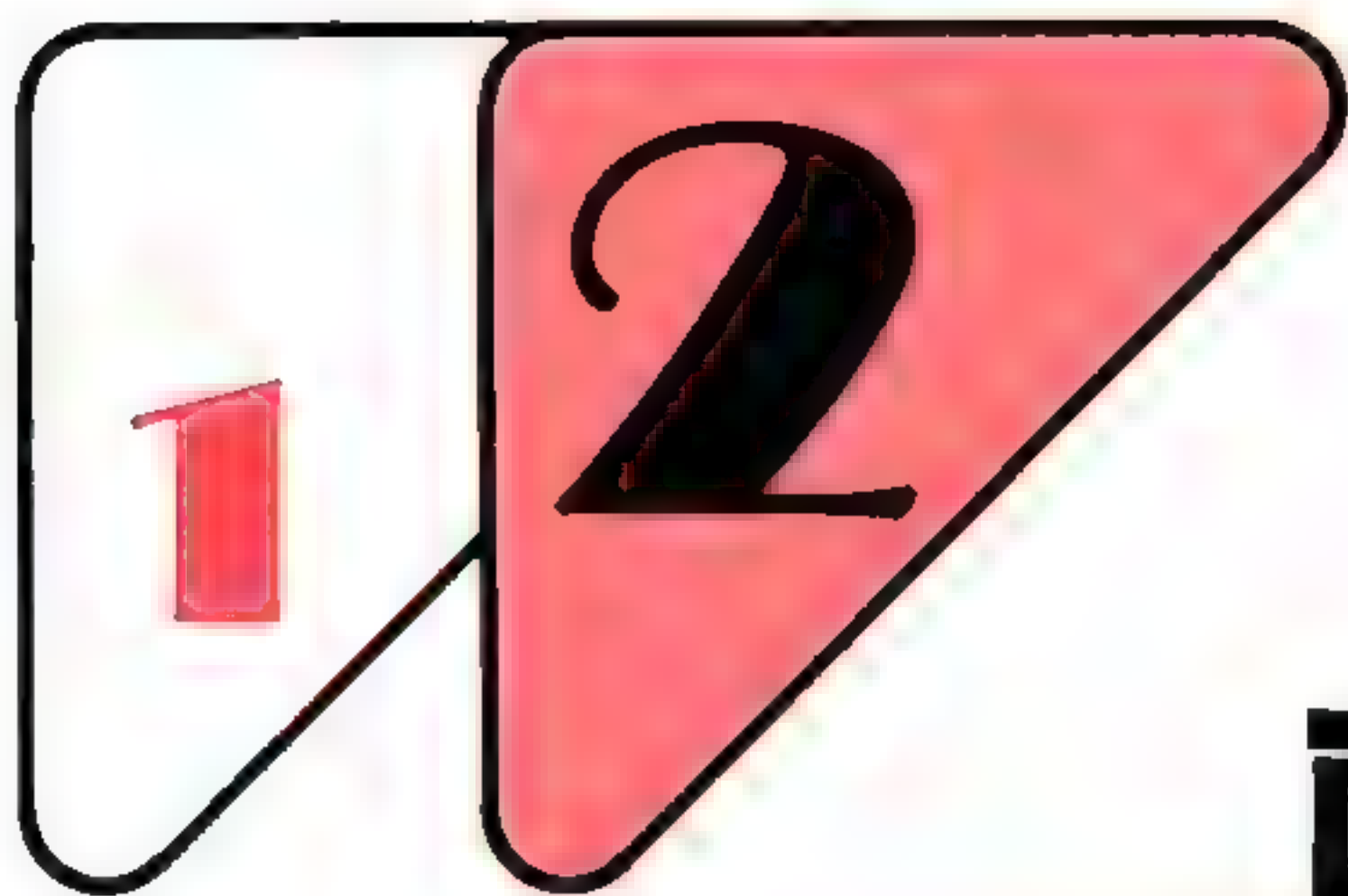
さぁ、答えを出してみよう

プログラムを実行させるときは[R][U][N][RETURN]キーだ

プログラムにつける番号を行番号というんだ

10番きざみにつけると便利だよ

● [RETURN] キーの役わりは？

キーボードから直接、命令を入力するときの[RETURN]キーの役わりは、「命令を実行しなさい」という意味だ。

プログラムを入力するときは、番号のついている1行ごとに[RETURN] キーを押さなければならないが、このときは「プログラムを記憶しなさい」という意味になる。「記憶したプログラムを実行しなさい」というときは、[R][U][N][RETURN]キーだ。

画面に文字を書く

表示するものを“　”で囲む

PRINT命令その2●文字やキャラクタを画面に表示する

PRINT命令には、もうひとつの働きがある。

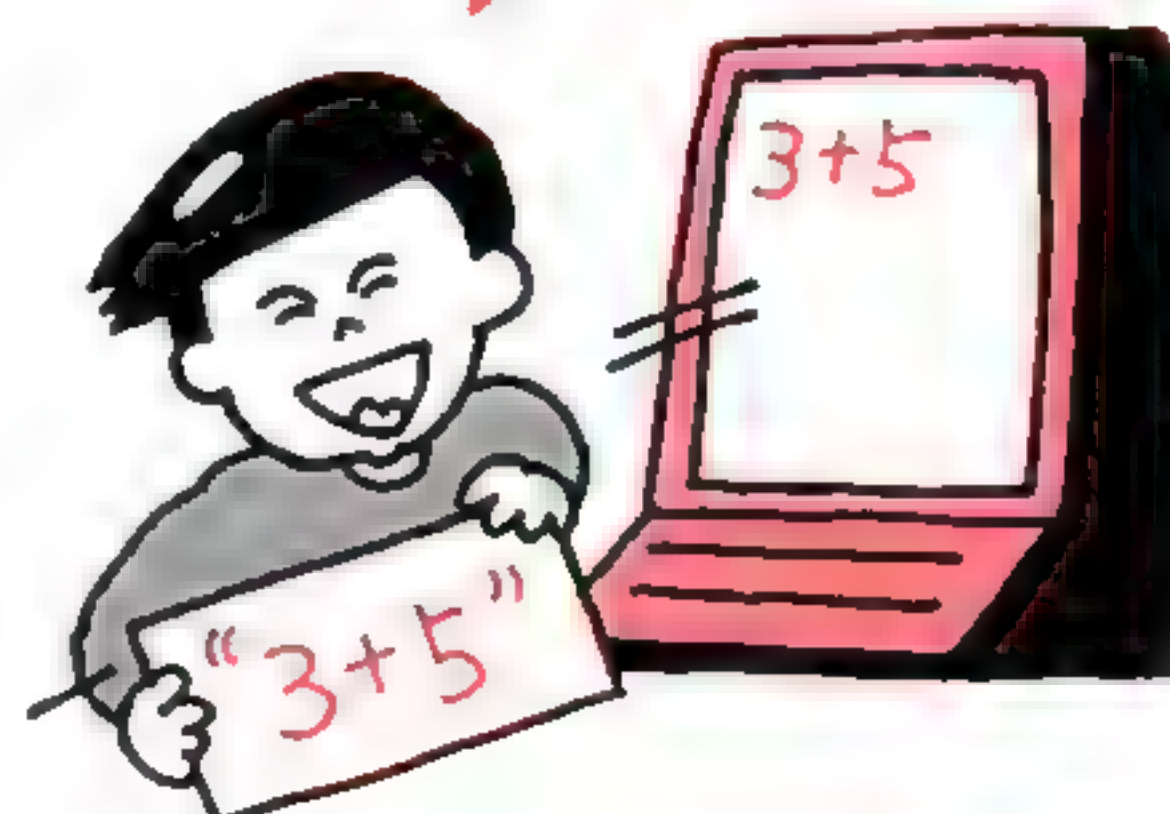
次のプログラムを入力してみよう。

```
10  CLS
20  PRINT  "3+5"
30  PRINT  3+5
```

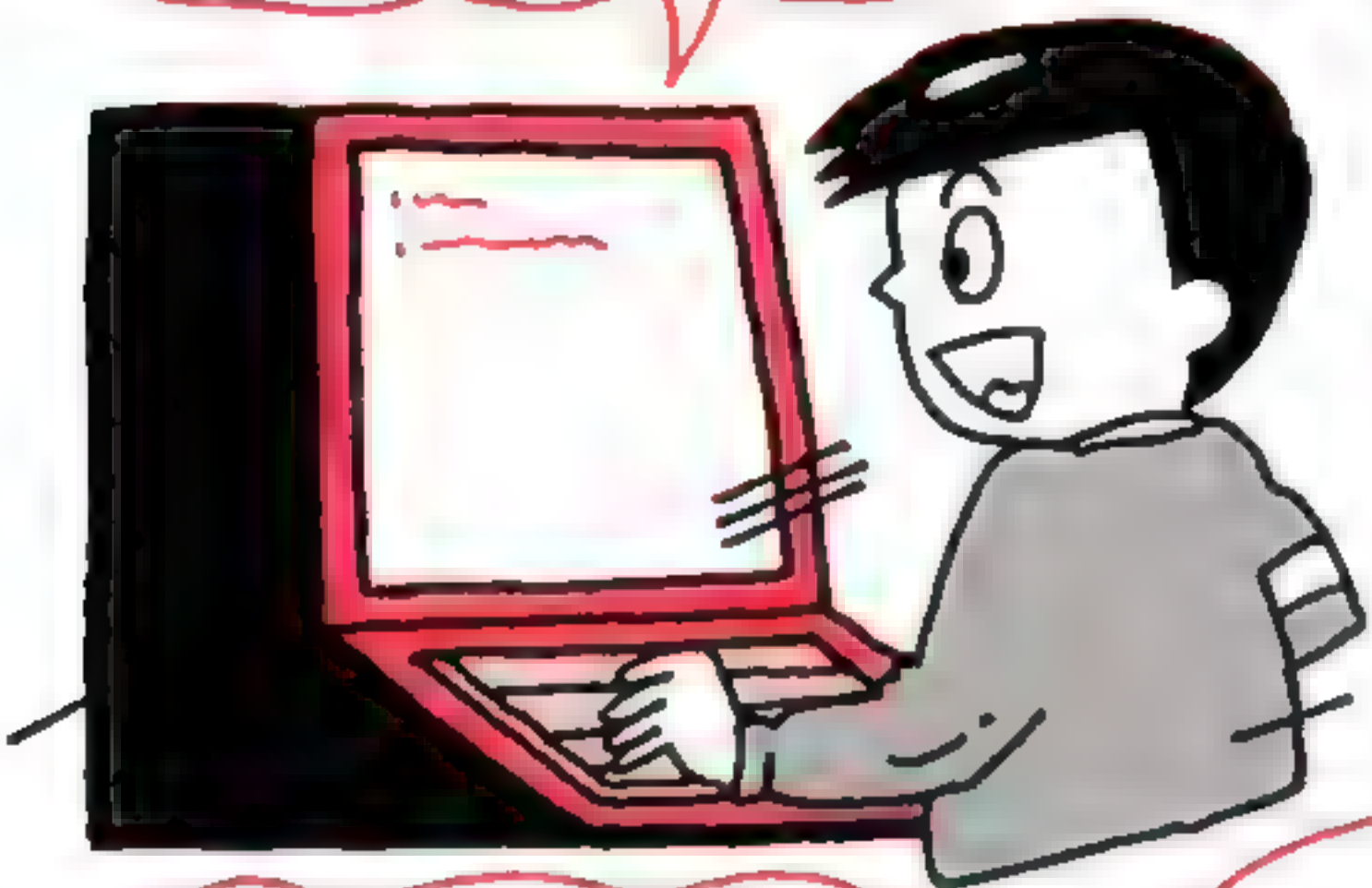
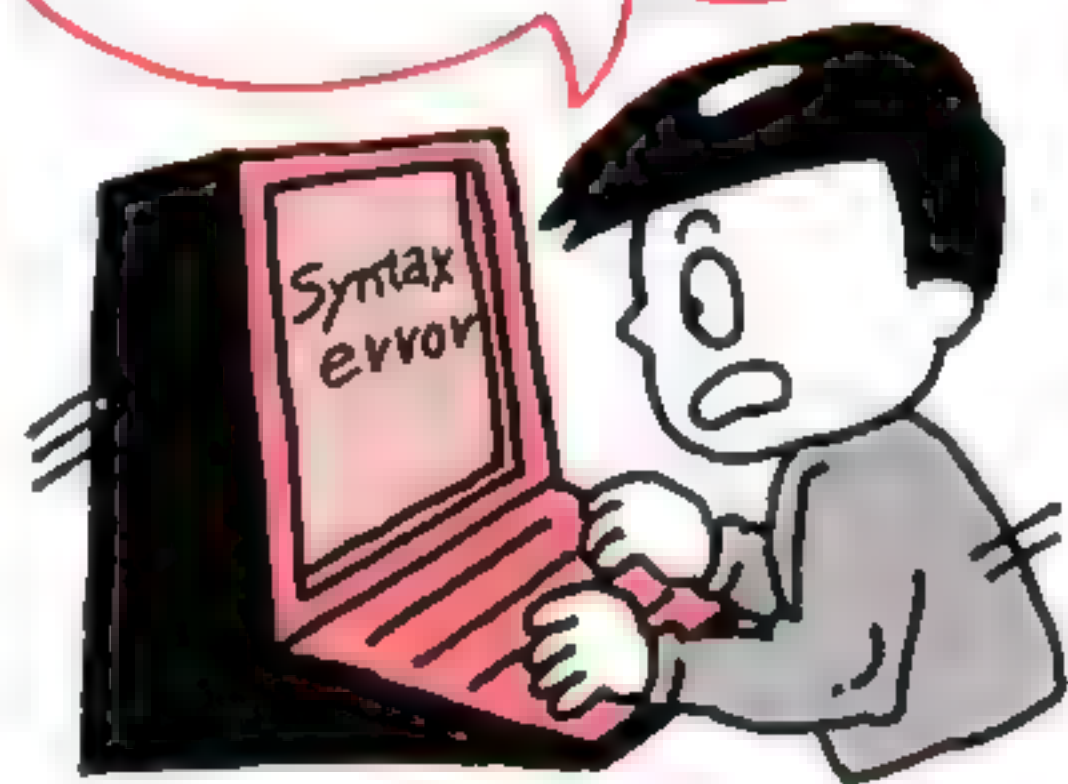
さあ、キー入力してRUNさせてみよう。画面はどうなっただろう。

```
3 + 5
8
OK
■
```

PRINTのあとの文字や数字を“　”（ダブルクォーテーション）で囲むと、「“　”の中を画面に表示しなさい」という意味になる。

だから20行のPRINT文では“　”の中を表示して、30行では計算をして画面に表示したんだ。

●アルファベットや模様を書こう

" 　 "（ダブルクォーテーション）つきのPRINT命令がわかったところで、右のプログラムを入力してみよう。空白のところはスペースキーを押すんだ

```
10 CLS
20 PRINT"***********"
30 PRINT"*         *"
40 PRINT"*         *"
50 PRINT"* ABCDEFG *"
60 PRINT"*         *"
70 PRINT"*         *"
80 PRINT"***********"
```

キー入力のときに、" 　 "を忘れないように注意しよう。もし忘れるとエラーが出てしまうぞ

エラーってな〜に？

あとでくわしく説明するけど、プログラムのまちがいや入力ミスをパソコンが教えてくれる

ＯＫ！　ぜんぶ入力できたよ。　R U N RETURNキーだね

あっ、出たぞ、出たぞ。＊の箱の中に、アルファベットが書いてある。これがPRINT命令なんだね

まちがったときはどうしよう

LIST命令●プログラムを画面に表示する

さて、プログラムを実行しても、ちゃんと画面に表示されなかった人はいないかな。

プログラムにミスがあると、パソコンは途中で実行が止まってしまう。そのかわりに、「ピー」という音を出して、プログラムのどこにまちがいがあるかを教えてくれる。

おや？　こんな英語の文が出ているぞ。

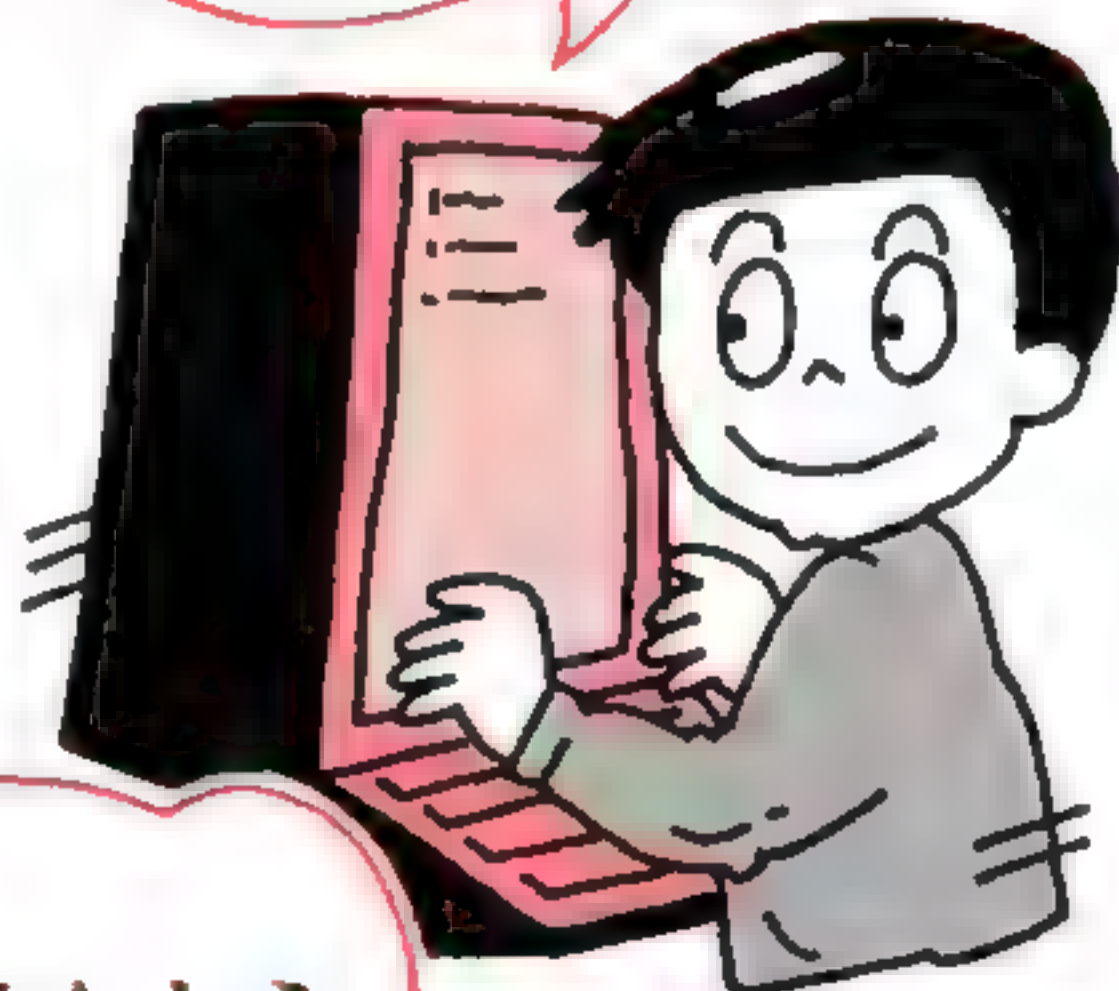
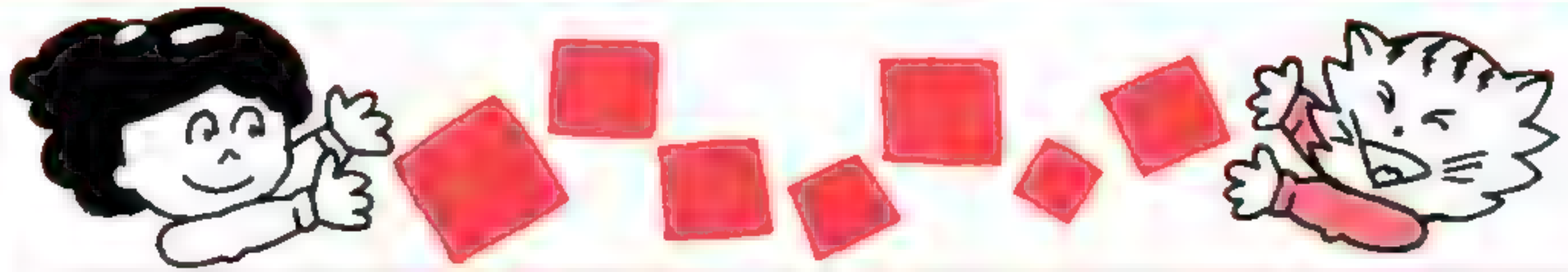
Syntax　error　in　50

これは、「プログラムの書きかたがBASIC文法に合っていません。50行にそのまちがいがあります」と、教えているエラーメッセージだ。

では、プログラムを直さなければいけないね。

入力したプログラムを、もう一度画面に表示してくれる命令が、LIST命令だ。

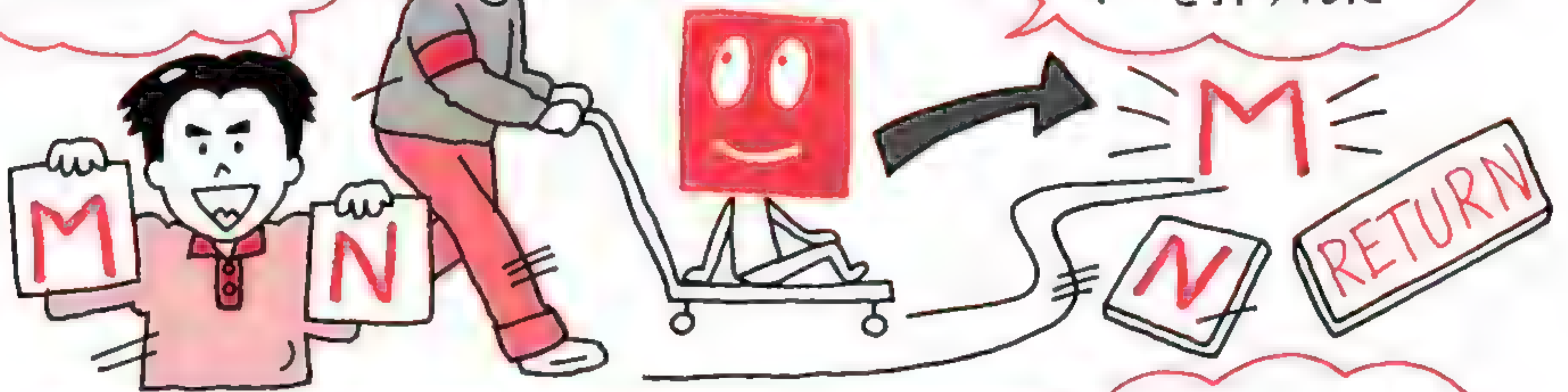
[L][I][S][T]とキー入力して、[RETURN]キーを押してみよう。

```
10 CLS
20 PRINT"***********"
30 PRINT"*         *"
40 PRINT"*         *"
50 PRIMT * ABCDEFG *"
60 PRINT"*         *"
70 PRINT"*         *"
80 PRINT"***********"
```

●打ちまちがえた字を正しい字に直す

50　PRIMT　＊　ABCDEFG　＊"

まず、PRIMTをPRINTに直してやろう

カーソルをここまで動かしてから

正しい文字Nを押して、RETURNキーを押すんだ

50　PRINT　＊　ABCDEFG　＊"

よし、直ったぞ　あとは、PRINTのあとに"を入れてやらなくちゃ

"がないと計算しちゃうけど、文字は計算できないから、エラーだ

LIST命令のいろいろな使い方

LIST　40…………………40行のプログラムだけを表示させる

LIST　10—40…………10行から40行までを表示させる

LIST　　—40…………最初から40行までを表示させる

LIST　40—……………40行から最後までを表示させる

LIST　……………………プログラムをぜんぶ表示させる

●文字や記号を追加する

文字を追加するときに使うキーは、INS（インサート）キーだ。

```
50 PRINT *  ABCDEFG  *"
```

追加したい字の右にカーソル移動だ

```
50 PRINT *  ABCDEFG  *"
```

INSキーを押すと、カーソルが2分の1の大きさになり、

カーソルの位置をまちがえないで

```
50 PRINT "*  ABCDEFG  *"
```

" をキー入力すると " が追加される

```
50 PRINT "*  ABCDEFG  *"
```

もう一度INSキーを押すと、カーソルが元にもどる

1　新しく文字を追加したい位置の右の字の上にカーソル■をかさねる。

2　INS（インサート）キーを押す。カーソル■が半分の大きさ▬になる。

3　追加したい文字のキーをたたく。文字が追加されて、そのぶん前からあった文字が右側に動く。

4　追加し終わったら、もう一度INS キーを押す。カーソルがもとの形■に戻る。最後に、RETURN キーを押す。

終わったら、もう一度LIST命令でプログラムを画面に出してみよう。ちゃんと直っているかな？

まちがえた文字を消す

まちがえた文字を消すときは、DEL（デリート）キーを使う。

```
10 PRINTT"ABCDEF"
```

カーソル■を消す文字の上にかさねる。

```
10 PRINT"ABCDEF"
```

DEL キーを押す。カーソル位置の文字が消える。

右側の字が1字分左に移動する。

```
10 PRINT"ABCDEF"
```

はじめのカーソル位置

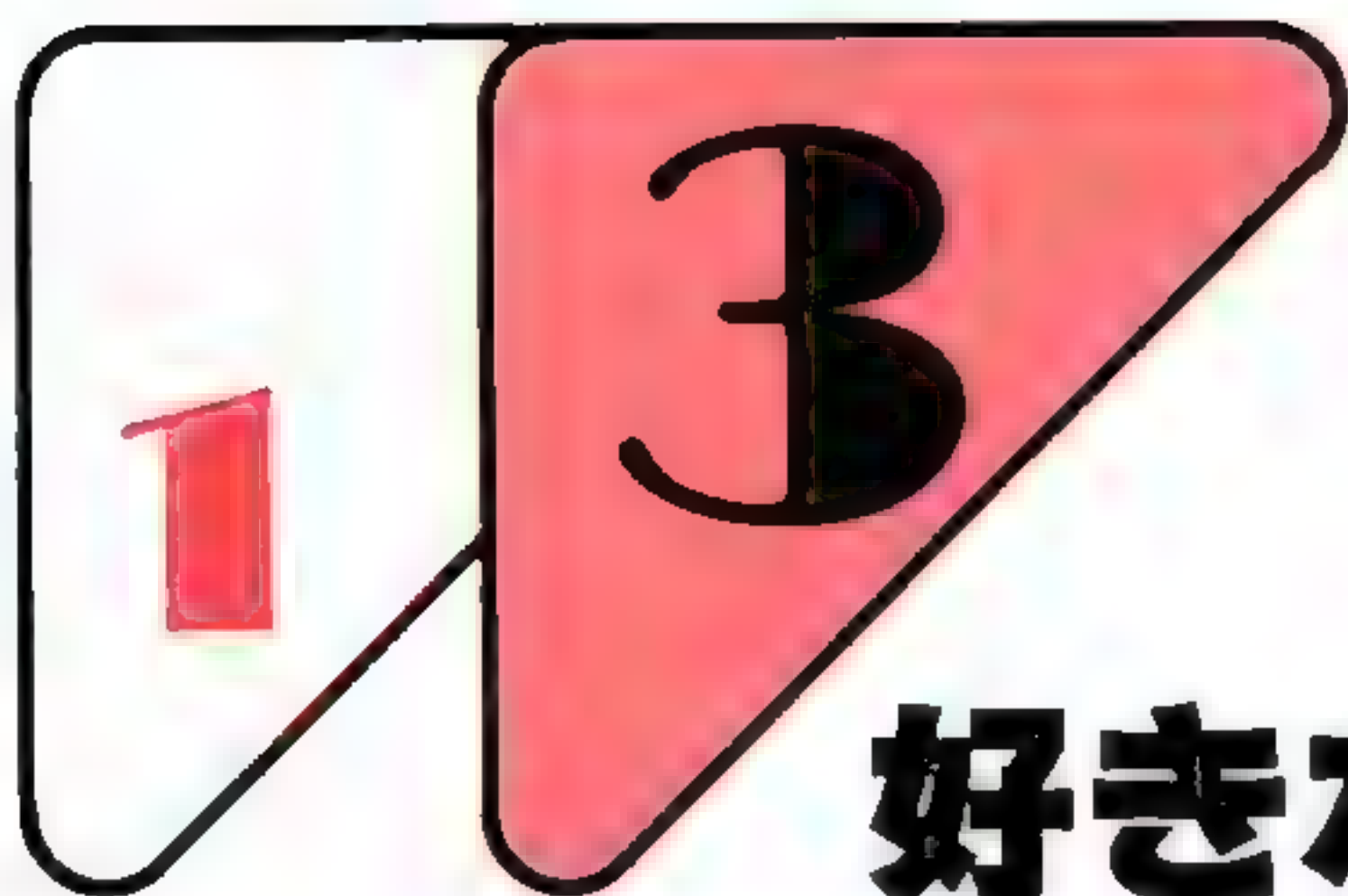
```
10 PRINT"AF"
```

1回押すと1文字を消しそのまま押し続けると押した回数だけの文字数が消去される。BCDEをまとめて消してみよう。

●画面表示中のリストを途中でストップするには？

短いプログラムであれば、一度にぜんぶのプログラムが画面に表示されるが、長いプログラムになると、上の1行から順番に消えて、下から続きのプログラムがどんどん表示されてくる。

この途中で画面を停止させたいときは、STOP（ストップ）キーを押して表示の実行を中断させる。もう一度 STOP キーを押せば、ふたたび表示が続けられる。

また、途中でLIST命令の実行を終わらせたいときには、CTRL キーと STOP キーを押す。

好きな位置に表示する

画面には29×24のマス目がある

LOCATE●画面の縦の位置と横の位置を指定する

さっきPRINT文で実行させた模様とアルファベットは、画面のどこに表示されたかな？　いちばん上の左はしから表示されたね。

では、画面のまん中とか下の方とか、自分の好きなところに表示させるにはどうしたらいいか。

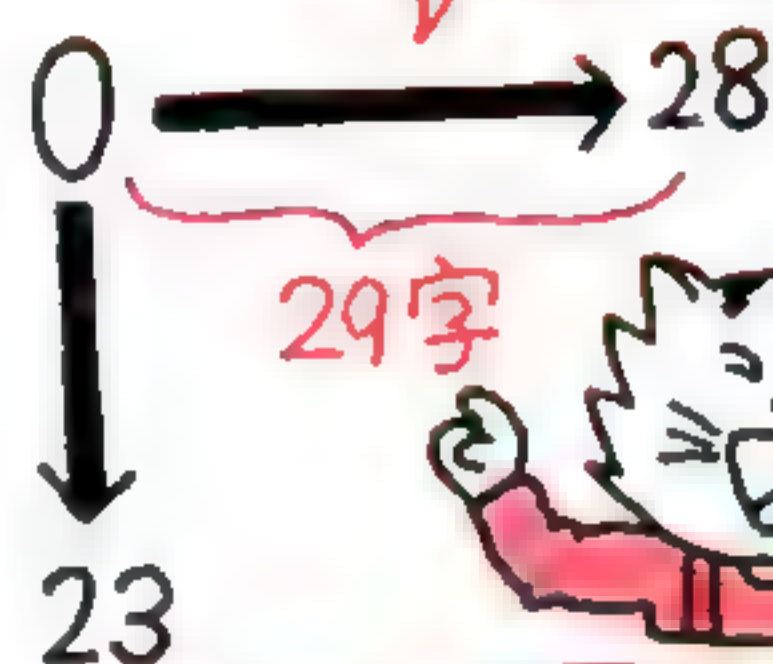
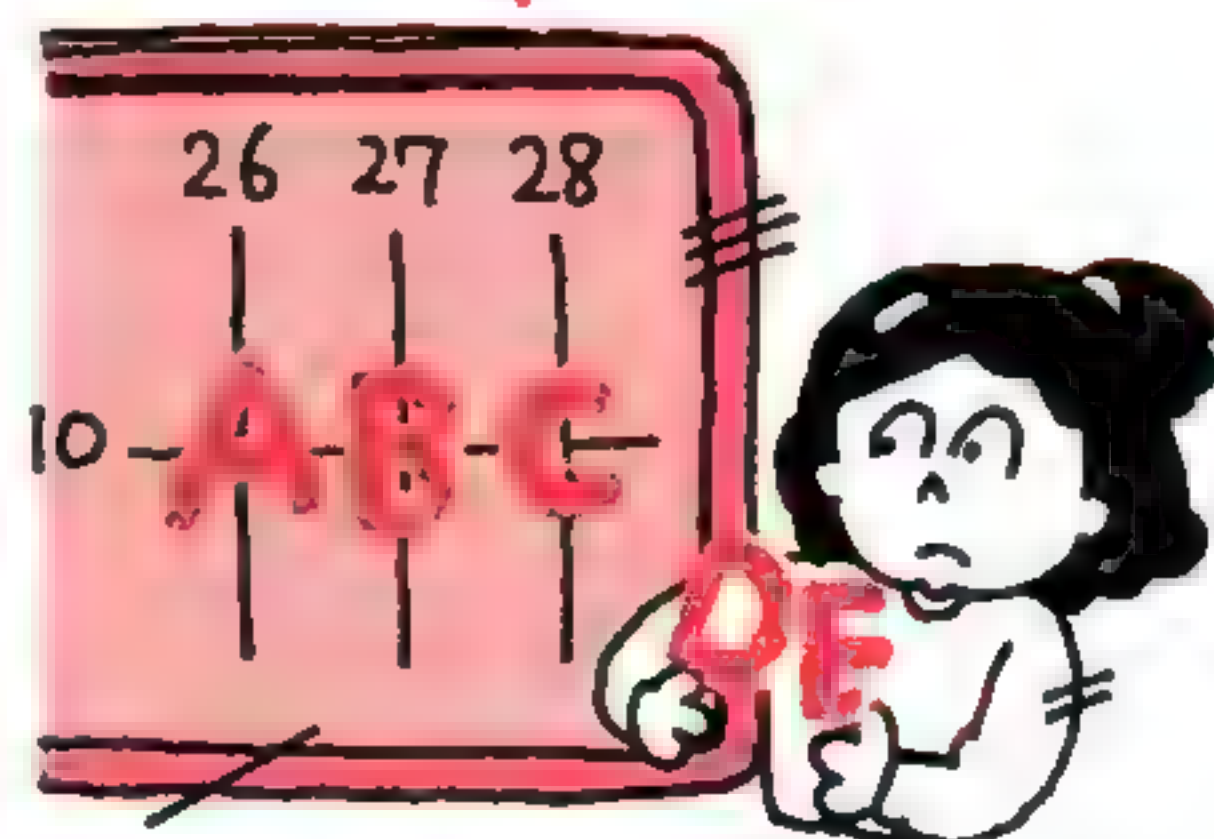
画面に表示する位置を決めるのが、LOCATE命令だ。

文字を書く画面は、横に29字、縦に24行の字が表示されるようになっている。位置を決めるために、横をX座標として0から28まで、縦をY座標として0から23までの番号が決められている。

この座標の位置をLOCATE命令で指定すると、その位置からはじめの1文字めが表示されるんだ。

次の式のX、Yにいろいろな数値を入れてためしてごらん。ただし、Xは0から28、Yは0から23までと決められていることに注意してね。

LOCATE　X座標, Y座標

指定した位置から1文字めが始まるんだから、文字数にも注意しよう

ABCDEを続けて表示させたいのにLOCATE　26, 10なんて指定すると、DEだけ次の行の頭から表示されてしまう

●ひとつのマス目に1字表示

```
10  CLS
20  LOCATE  10, 5
30  PRINT  "ABC"
40  END
```

Aの字がどこから表示されるか見てみよう

文字や記号を表示させる画面がテキストモードだ。画面の縦、横の番号は0から始まることに注意しよう。

次のプログラムを実行してみよう。

```
10 CLS
20 LOCATE 2,3:PRINT "MSX"
30 LOCATE 10,10:PRINT"♥♥♥♥♥♥♥♥"
40 LOCATE 10,11:PRINT"♥      ♥"
50 LOCATE 10,12:PRINT"♥      ♥"
60 LOCATE 10,13:PRINT"♥      ♥"
80 LOCATE 10,14:PRINT"♥♥♥♥♥♥♥♥"
90 LOCATE 12,12:PRINT "MSX"
```

上のプログラムの意味はわかったかな？　20行で横3マスめ、縦4マスめのところから「MSX」を表示する。そのあと、♥で箱を書き、その中にMSX（90行）を表示する。こうして、グラフィックキャラクタを使えば、かんたんな模様なども表示させることができるよ。

●グラフィックキャラクタを表示するプログラム

グラフィックキャラクタを使って、トランプの「スペードの3」と「戦車」を表示させてみた。

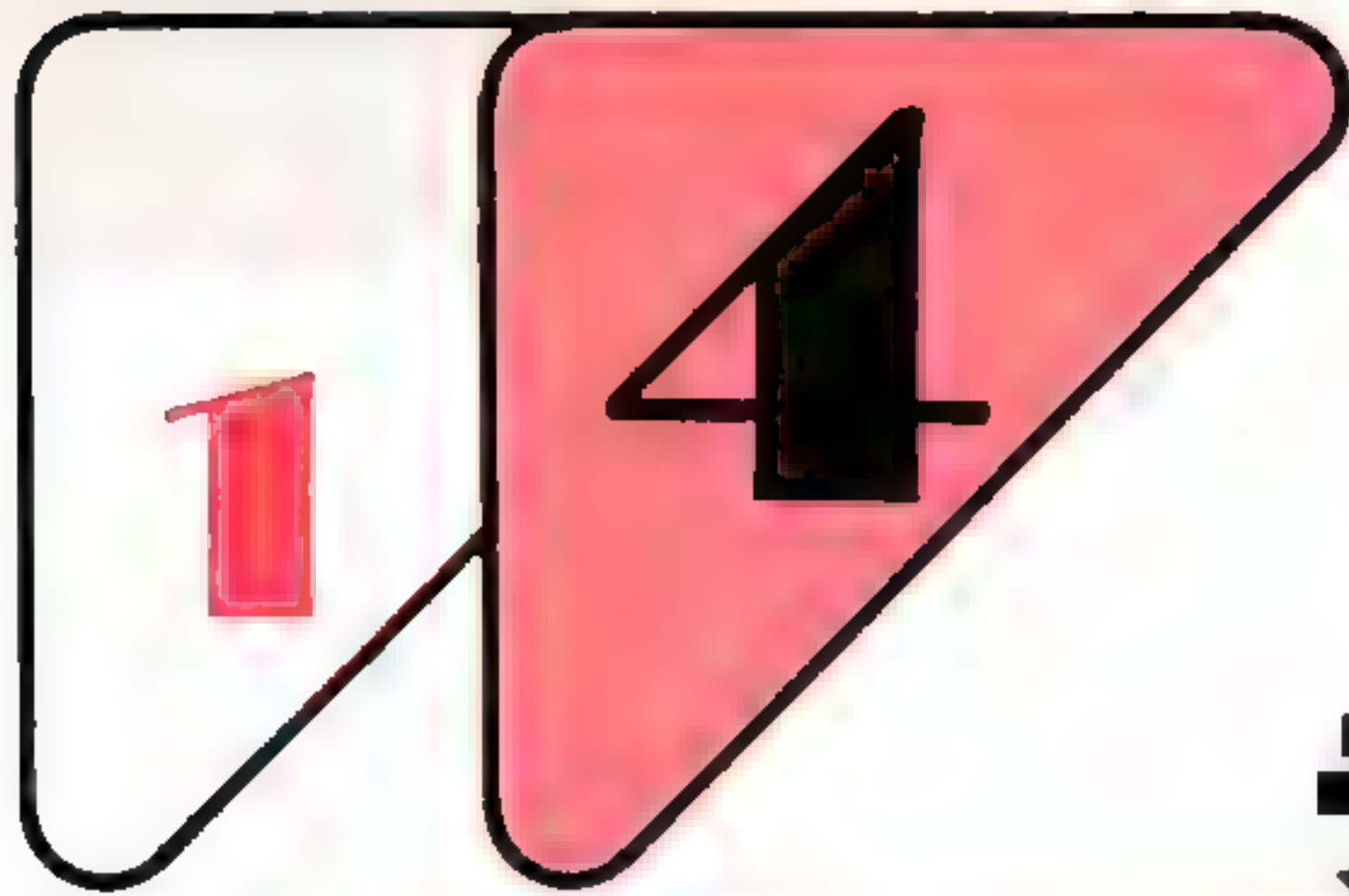
「スペードの３」を表示する

```
10 CLS
20 LOCATE 3,3:PRINT"┌────┐"
30 LOCATE 3,4:PRINT"|♠   |"
40 LOCATE 3,5:PRINT"|  ♠ |"
50 LOCATE 3,6:PRINT"|   ♠|"
60 LOCATE 3,7:PRINT"└────┘"
70 END
```

「戦車」を表示する

```
10 CLS
20 LOCATE 10,10
30 PRINT"--[#]"
40 LOCATE 10,11
50 PRINT"  [#]"
60 LOCATE 10,12
70 PRINT"<*****>"
```

命令と命令をつなぐ：（コロン）の役わり

左のプログラムは、ひとつの行に、LOCATEとPRINTがいっしょに並んでいて、それぞれの命令が：（コロン）で区切られている。：を使うと、ひとつの行の中にいくつもの命令をつなげて書くことができる。これをマルチステートメントというが、あまり使いすぎると見にくくなるので注意しよう。

データを入れる箱

数を入れる箱 “数値変数”

=(イコール記号)は「等しい」ではなく「代入する」の意味

「変数」は、命令とは違うけれど、プログラムを組むうえで、たいせつな考え方のひとつだ。

次の計算問題をやってみよう。

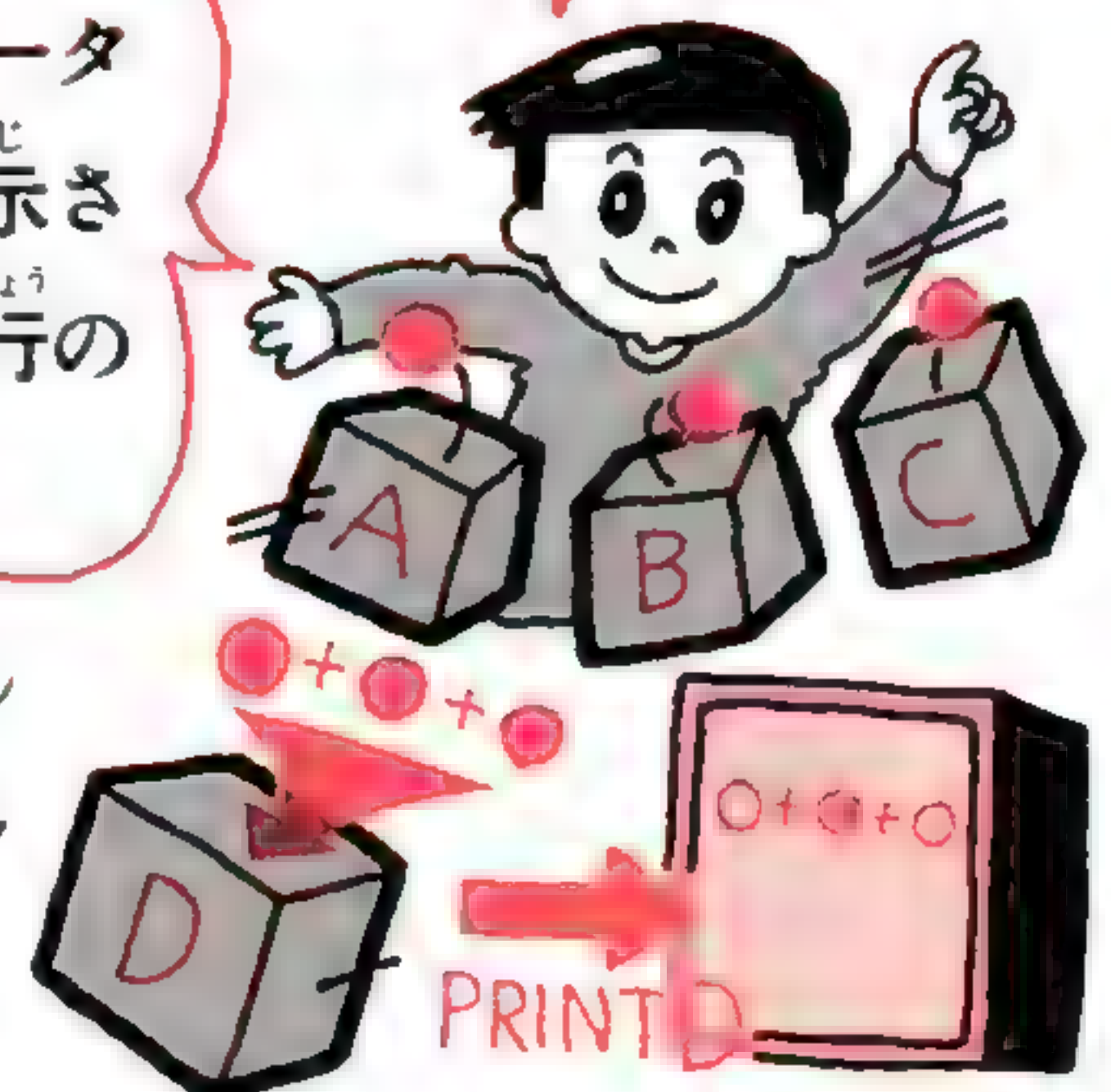
> Aの値は10、Bの値は20、Cの値は30です。
> D＝A＋B＋CのDの値はいくつでしょう。

かんたんな問題だね。AとBとCの値をたした答えDは60だ。

このA、B、C、Dが、コンピュータのプログラムの中で「変数」とよばれるものなんだ。

この計算問題をプログラムに直してみよう。

```
10 CLS
20 A=10:B=20:C=30
30 D=A+B+C
40 PRINT D
```

算数の計算では、＝（イコール記号）は「等しい」という意味をあらわすけれど、コンピュータでは「代入する」という意味なんだ。

同じ箱にデータが入ったり出たり

プログラム中の＝が代入する意味だということがわかるプログラムをもうひとつあげておこう。

```
10 CLS
20 A=5:B=8:C=10
30 D=A*B:A=C/2
40 PRINT D,A
```

30行では、２つの計算式を：（コロン）でつなげているよ

40行の，（コンマ）は、Dの値とAの値をはなして表示させるための，だ

まず自分で変数A、B、C、Dそれぞれの値を考えてから、プログラムをRUNさせよう。きみの考えた数値と、画面に表示された値が一致していれば、きみは変数がわかったということだよ。

変数名と変数値

ＡやＢが変数名で、変数に代入される値が変数値だ。

変数にも名まえをつけてやらないと、コンピュータが判断できない。変数名をつけるには、いくつかの約束ごとがあるので注意しよう。

1　アルファベットのＡ～Ｚ、数字の０～９までが使える。

2　数字だけの変数名はつけられない。

3　最初の１文字はアルファベットを使う。

4　２文字以内でつける

5　大文字でも小文字でもよい。

Ａ１はＯＫ、１Ａはダメ

BASIC命令にあるIFやTOは使えないよ

●変数は変化する数、Aの値はどんどん変わる

20行まではわかるね。

さぁ、問題は30行だ。

2つの計算式がある。

D＝A＊B

A＝C／2

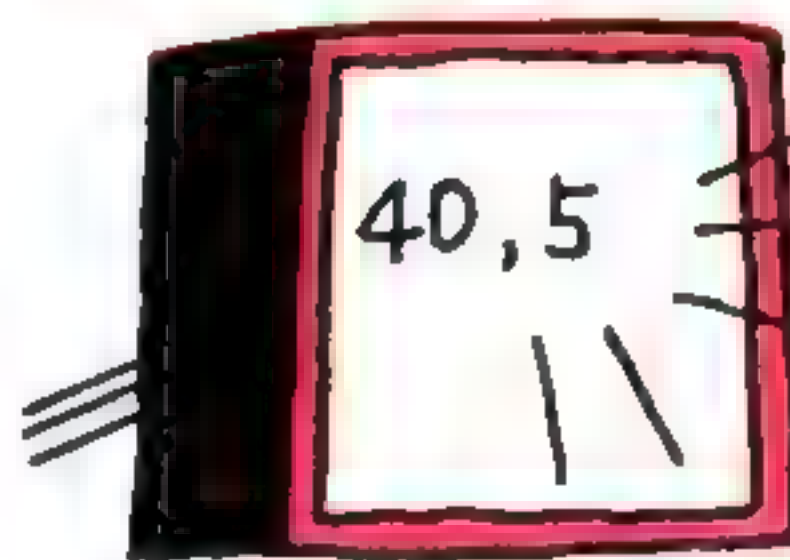
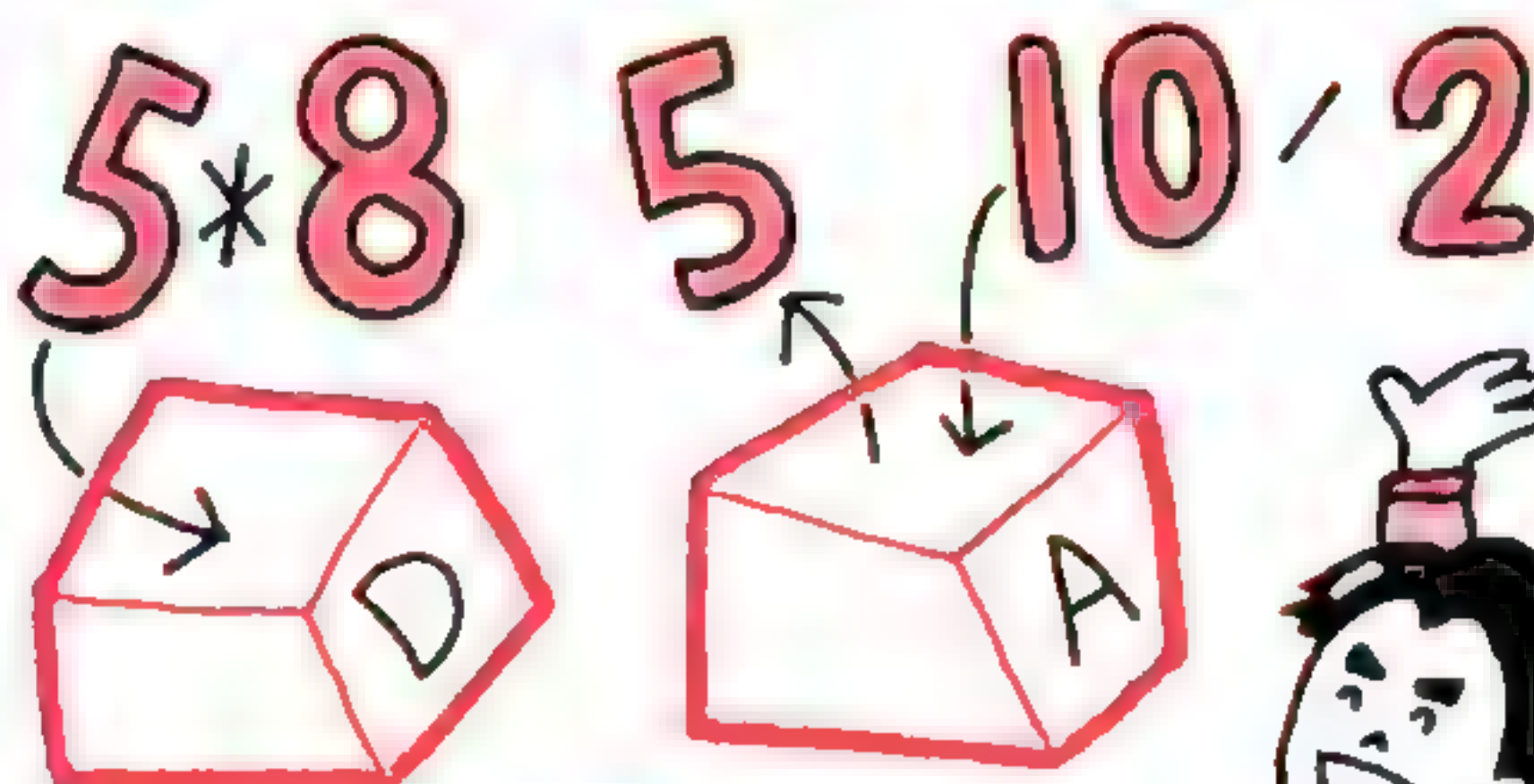
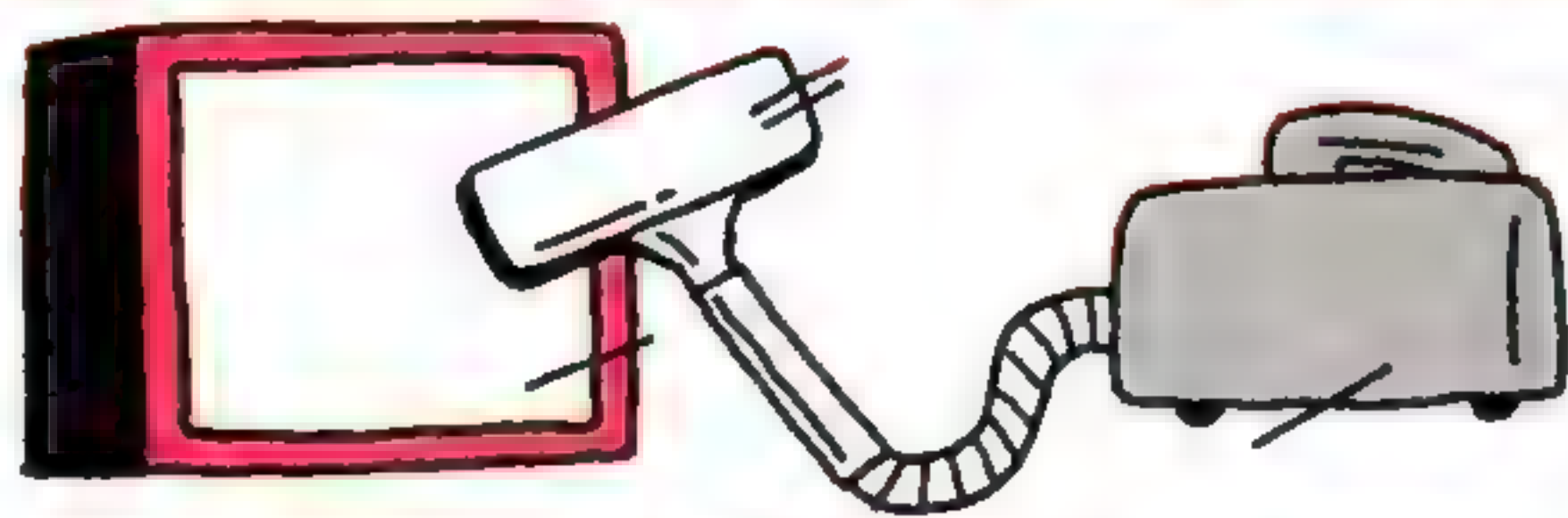
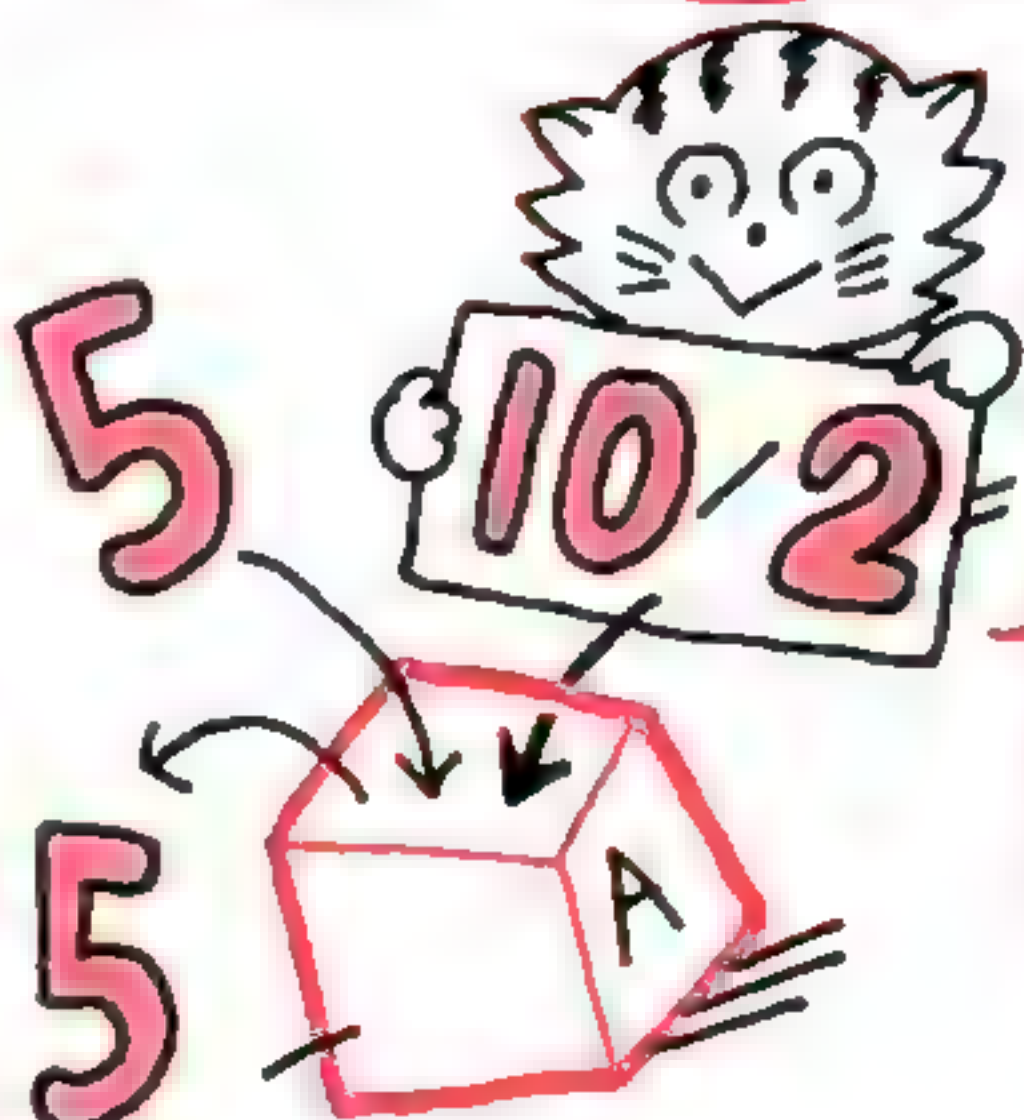
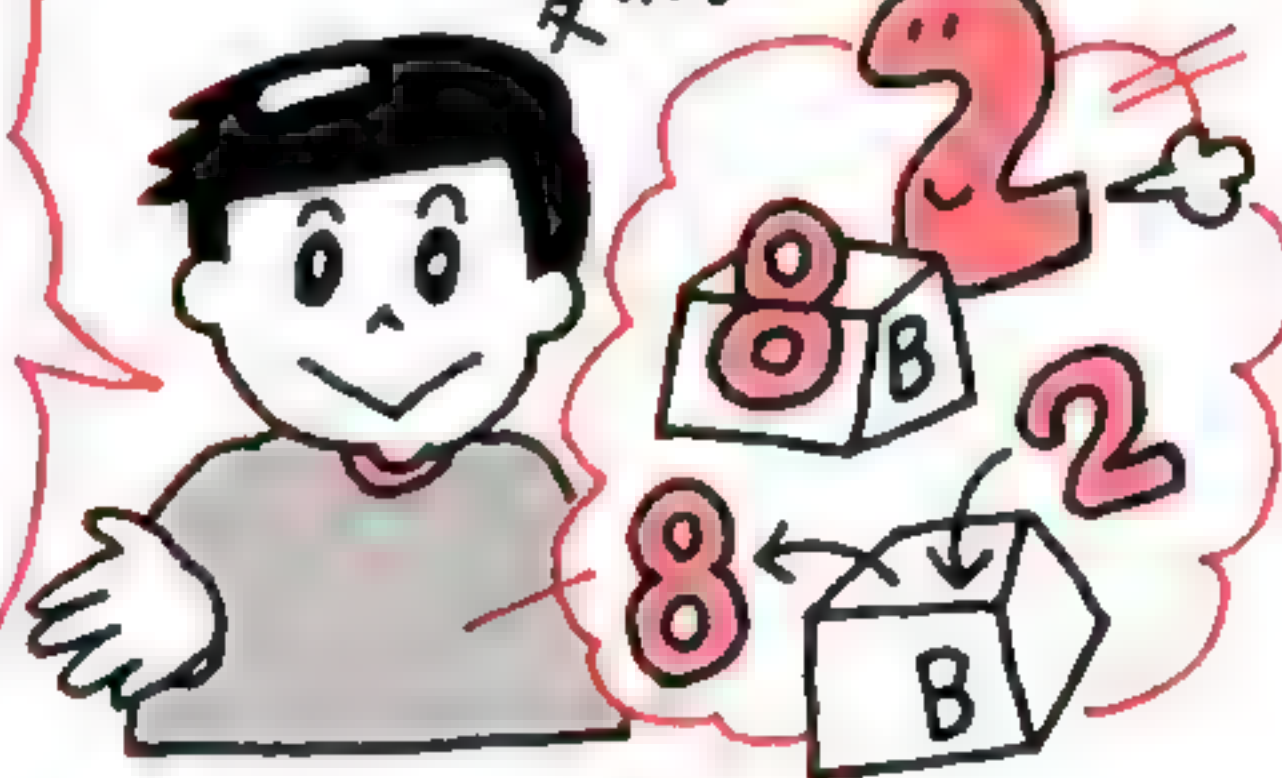
Dの箱には、A(5)とB(8)をかけた数を入れる。

次のAは？　20行の5を入れて

5＝C／2

なんてやった人はいないかな？

A＝C／2は、Aの箱にC／2の値を入れるという意味だけど、そうすると、前にAの箱に入っていた5はどうなったのだろう。

同じ変数に新しいデータを入れると、古いデータは出てしまうんだ。つまり、5は出てしまい、C／2が入る。

このプログラムでは、B、C、Dの箱(変数)は一度しか使わないから変数値は変わらない

Aの箱には、データが出たり入ったりして、最後に入っているデータが変数値になる

文字のデータを入れるときは

文字変数は変数名のうしろに＄(ドルマーク)をつける

データには大きくわけて２つの種類がある。ひとつは数値データ、もうひとつはことばなどの文字データだ。

数値データを入れる箱を、「数値変数」といい、いま計算問題で使った変数だ。

文字データを入れる箱は、「文字変数」または「ストリング変数」という。名まえはおぼえなくてもいいけれど、次の２つのことを守らなければいけない。

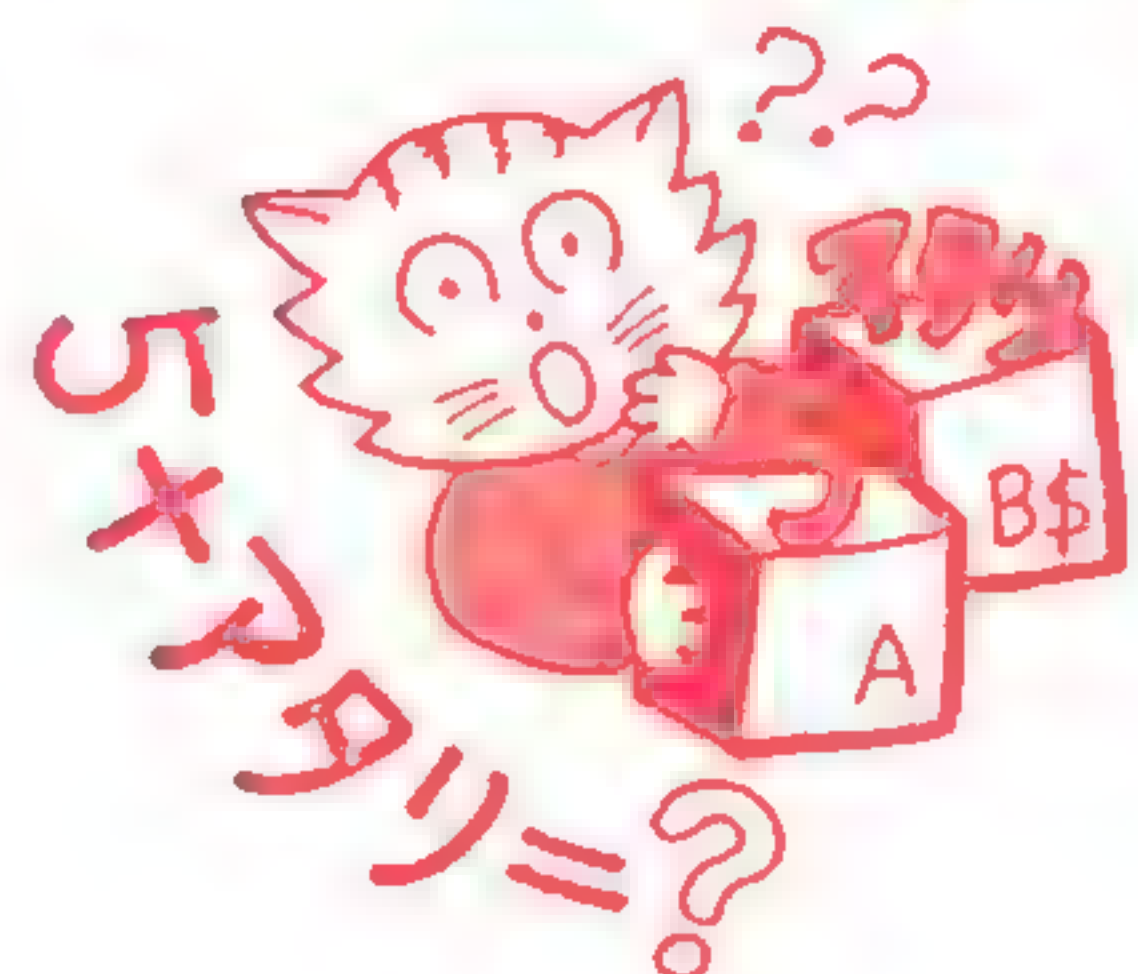
1　文字変数には＄（ドルマーク）をつけて、数値変数と区別する。

2　文字変数に入れるデータはかならず〝　〟（ダブルクォーテーション）で囲む。

たとえば、

```
10　CLS
20　A＄＝〝アタリ！〟
30　PRINT　A＄
```

文字変数に数を入れると、コンピュータは数値ではなく、文字だと判断されることに注意しよう。５は数値だけど、〝５〟はただの文字の５だ。

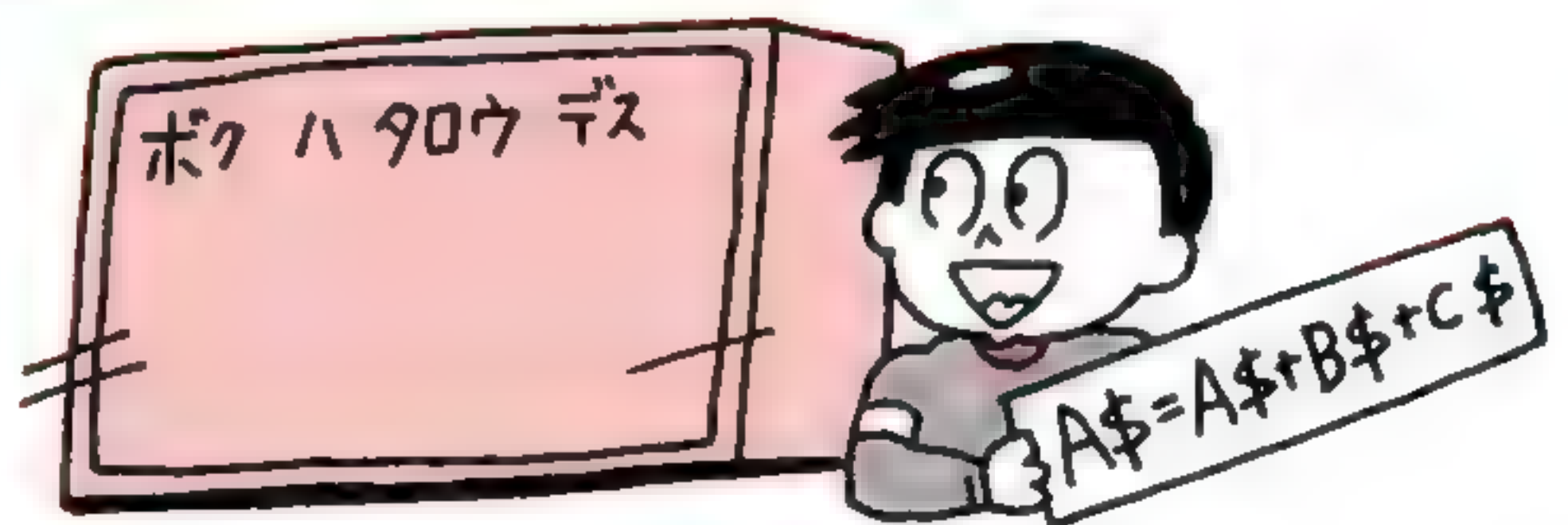
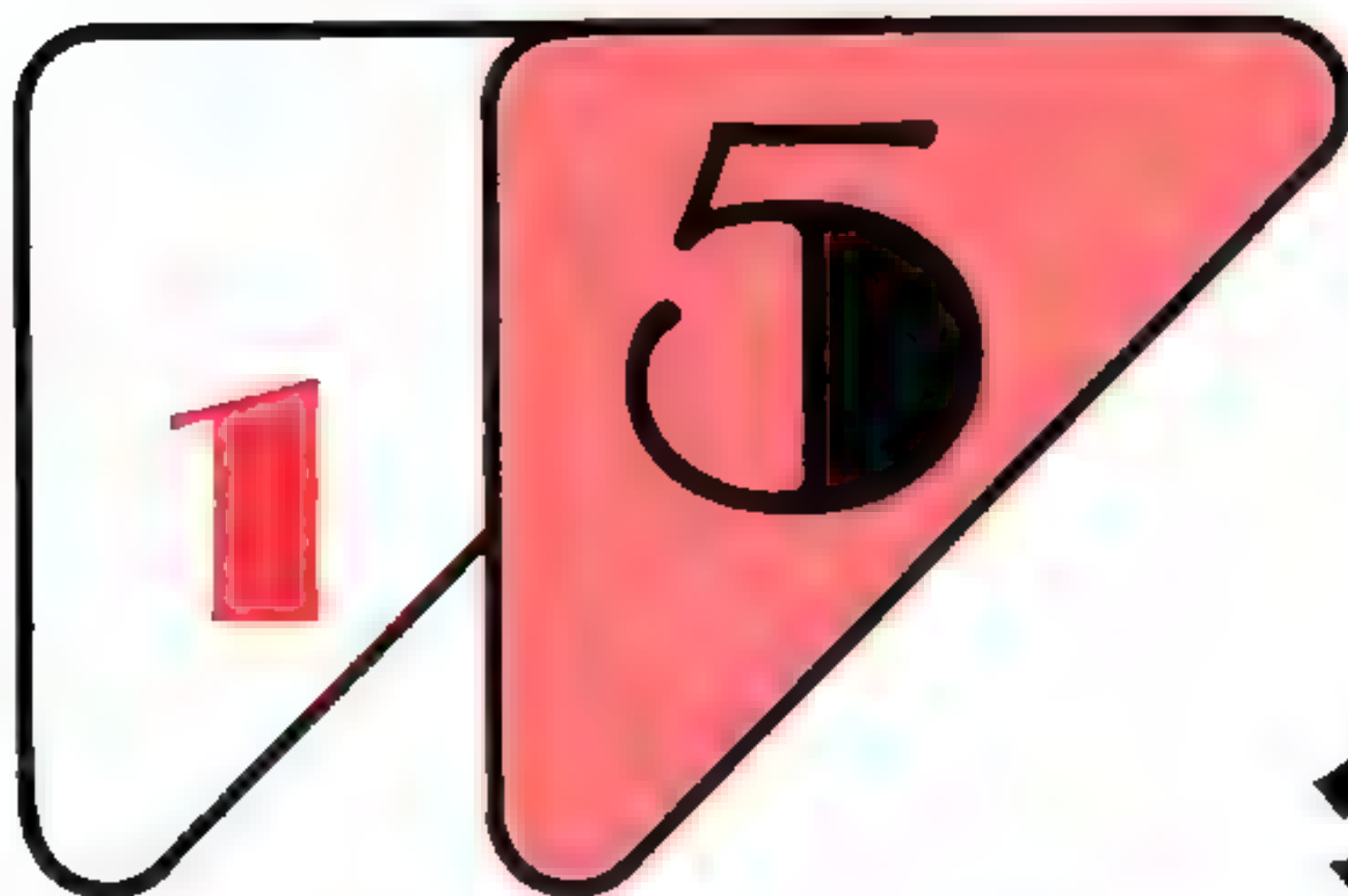
文字変数もたし算ができるけど、計算するのではなく、文字列と文字列をつなぐんだ。

●文字変数に代入された数字は数ではなく文字列だ

```
10 CLS
20 A=5:B=7
30 A=A+B
40 PRINT A
50 END
```

```
10 CLS
20 A$="5":B$="7"
30 A$=A$+B$
40 PRINT A$
50 END
```

数値変数と文字変数のちがいを見てみよう。

変数A（数値）　　変数A$（文字）

```
10 CLS
20 A$="ボク ハ "
30 B$="タロウ "
40 C$="デス"
50 A$=A$+B$+C$
60 PRINT A$
70 END
```

文字変数のたし算は、数値変数のように計算するのではなく、変数に代入した文字と文字をつなぐことだ。もちろん、文字だから、ひいたり、かけたり、わったりはできない。

クルマを走らせる

表示の位置をどんどん変える

●LOCATE、PRINT、変数のまとめプログラム

```
10 CLS
20 SCREEN 0,0
30 WIDTH 40
40 FOR N=1 TO 35
50 LOCATE N,10:PRINT" ┌─┐ "
60 LOCATE N,11:PRINT"┌┘ └┐"
70 LOCATE N,12:PRINT"└●─●┘"
80 LOCATE N-1,11:PRINT" "
90 LOCATE N-1,12:PRINT"*"
100 NEXT
```

今まで勉強した命令を使って、画面にクルマを走らせてみよう。

クルマは、LOCATE命令で表示する位置を決め、PRINT文で─と●を組み合わせて表示させる。これが50行から70行までのプログラムだ。

80行の" "の中にはなにも書かれていない。これは、車が移動したあとに残る影を消しているんだ。90行の＊で、けむりのかんじを出してみた。

横に走らせるということは、画面の横の座標、つまりX座標の上を移動させるということだね。上下に動かすならY座標の上ということだ。

20行のSCREEN命令は、文字や記号を表示するためのテキストモードを指定しているんだ

ふつうテキストモードのX座標は29文字表示できるようになっているけど、WIDTH 40を指定すると40文字に広がるんだ。それが30行だ

このLOCATE命令にあるX座標のNが、クルマを走らせるポイントなんだ。

Nは変数だから、そこにX座標の値を入れてやらなければいけない。40行を見てみよう。

FOR　N＝1　TO　35

これは、「Nに1から35までの値を入れよ」という意味なんだ。つまり、いちばんはじめのクルマは、50行から90行にあるNに1の値を入れたところに表示される。次のクルマは、Nに2を入れたところに表示される。次は、3、4………と最後にNに35を入れたところに表示される。

これを続けて表示させれば、画面の左から右へクルマを走らせることができるわけだ。

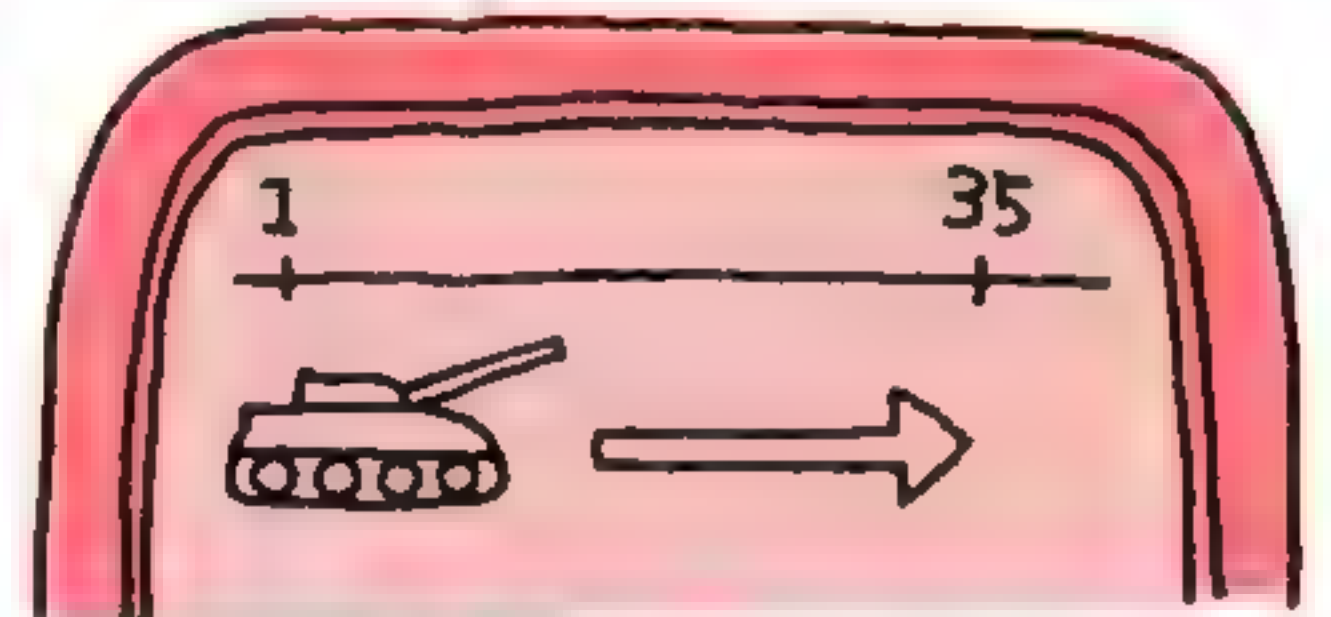
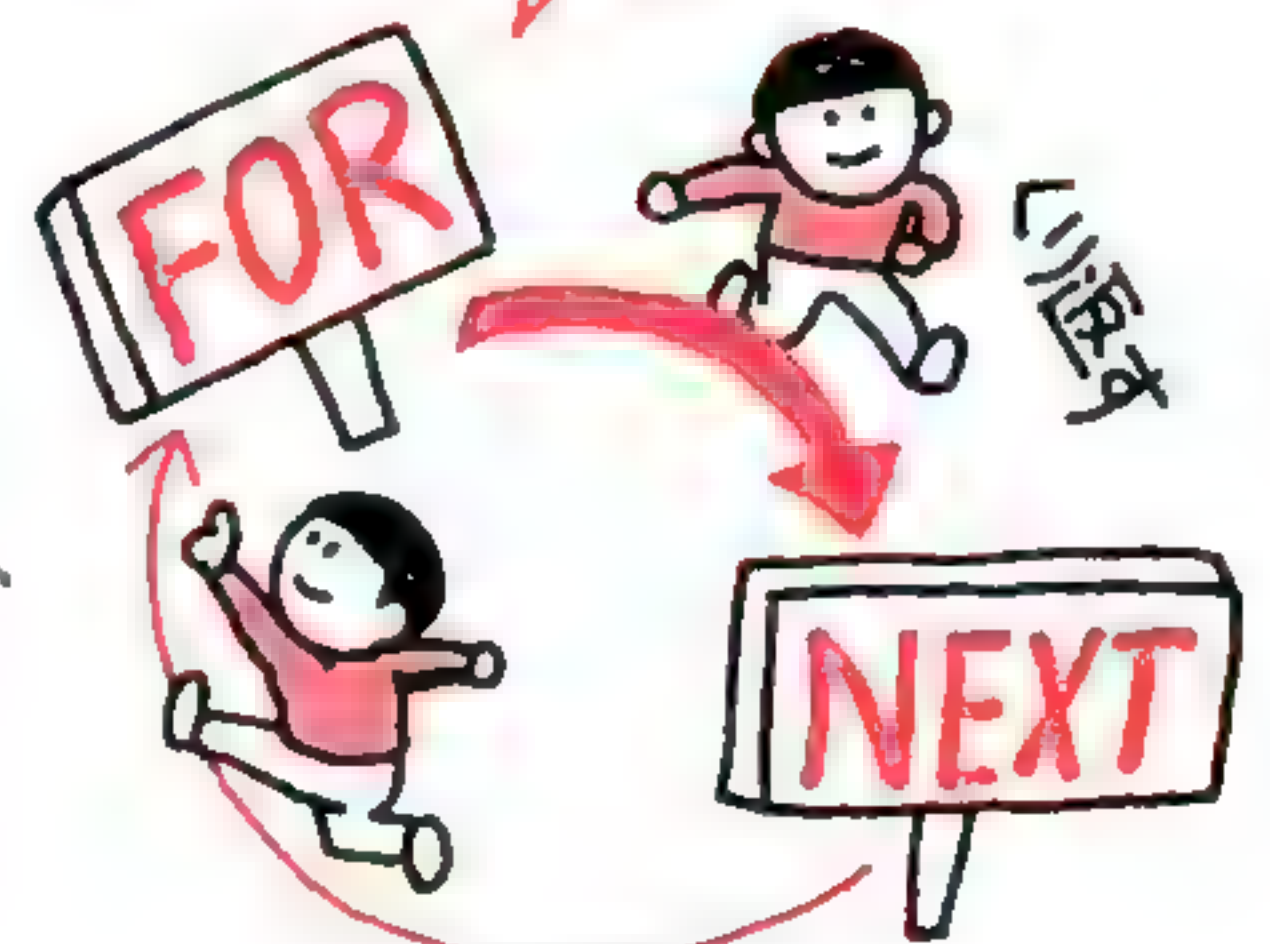
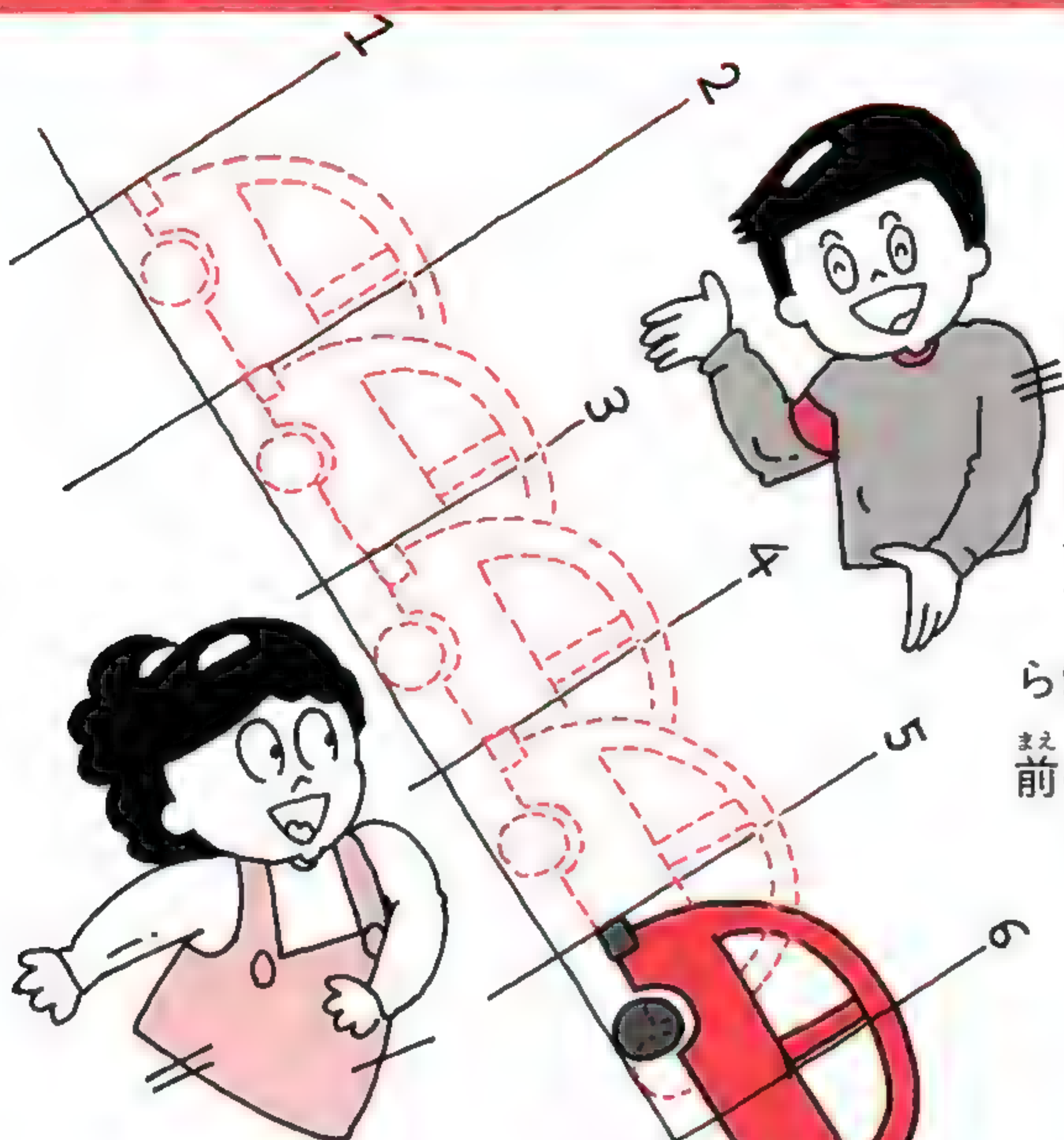
クルマの位置より1つ左にけむりを表示させるのだから、けむりの位置はN－1になるね

前に作った戦車をこのプログラムのように動かしてみよう

FOR N=1 TO 35

●FORとNEXTのあいだを35になるまでくり返す

X座標の上を1から35まで移動するのだから、Xの値を1から35まで変えて、35個のクルマを書くプログラムを作ってRUNさせれば、走らせることができる。でもそれでは、とても長いプログラムになってしまうね。

これを解決してくれるのが、FOR～NEXT命令なんだ。NEXTは100行にあるね。FORとNEXTにはさまれた命令を、決められた回数だけくり返すんだ。このプログラムでは、Nが35になるまで、「クルマを表示する」という同じ仕事をくり返してくれる。ずいぶん便利な命令だね。

FOR～NEXTのくわしい説明が68ページにあるから、ちょっと読んでみてごらん

クルマが画面の上を走る、走る、走る

最初のクルマの位置は、

Nに1を入れて、

```
LOCATE 1, 10
LOCATE 1, 11
LOCATE 1, 12
```

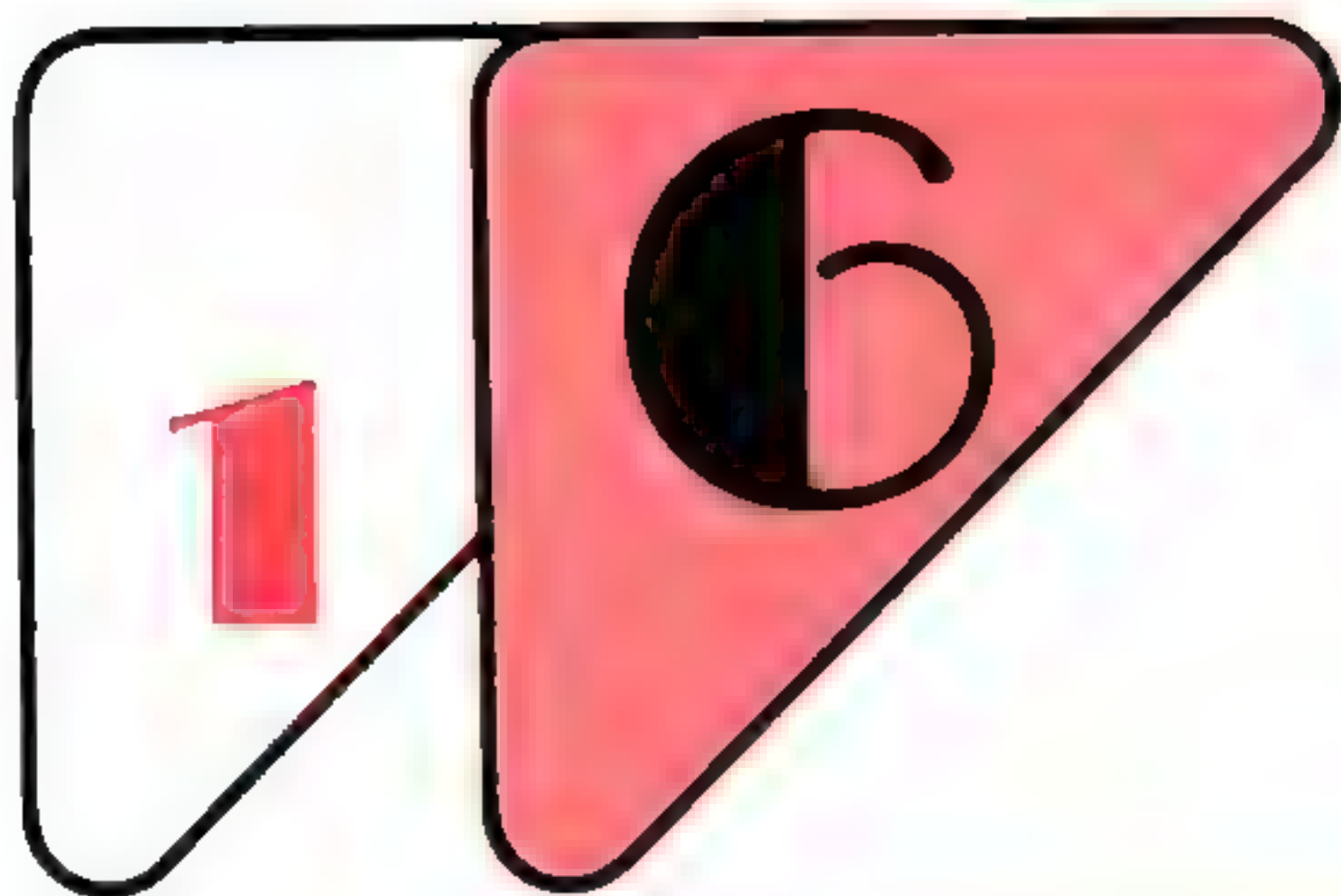
次はNに2を入れて、50行から90行のプログラムをくり返す。前のクルマは消える。

```
LOCATE 2, 10
LOCATE 2, 11
LOCATE 2, 12
```

その次は、Nに3を入れる

その次は4

Nに35が入るまで同じ仕事をくり返す。このくり返しの仕事をしてくれるのが、FOR～NEXT命令だ。

いま表示したクルマを消して、その位置の右どなりに新しいクルマを表示させる。この動作をくり返すと、見た目では、クルマがどんどん右に走っているように見えるんだ

プログラムを入力する前に

メモリ、行番号、表示文字数

NEW命令●メモリの中のプログラムを消してスッキリメモリ

NEW命令は、メモリの中に入っているプログラムを消してやる命令だ。

スイッチを入れたまま、次から次へとプログラムを入力していくと、メモリの中はどんどんいっぱいになって、いろいろなプログラムがまざり合ってしまう。そうすると、新しく入力したプログラムが、正しく実行されないことがあるんだ。

だから、スイッチを入れたままのときは、次のプログラムを入力する前に、[N][E][W][RETURN]キーで、記憶されているプログラムを消去したほうがいいよ。

LIST命令を実行してみて、画面にプログラムがなにも表示されなければ、NEW命令がちゃんと実行されているよ

AUTO命令●行番号を自動的につけてくれる

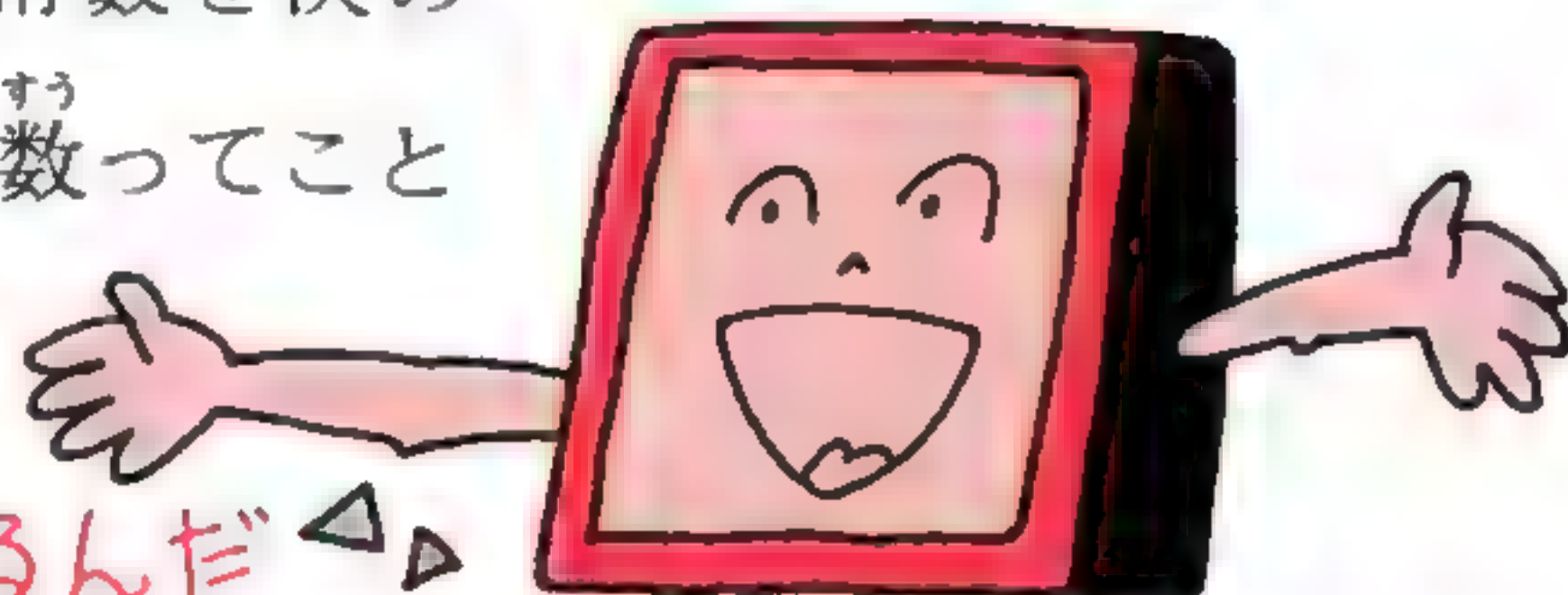
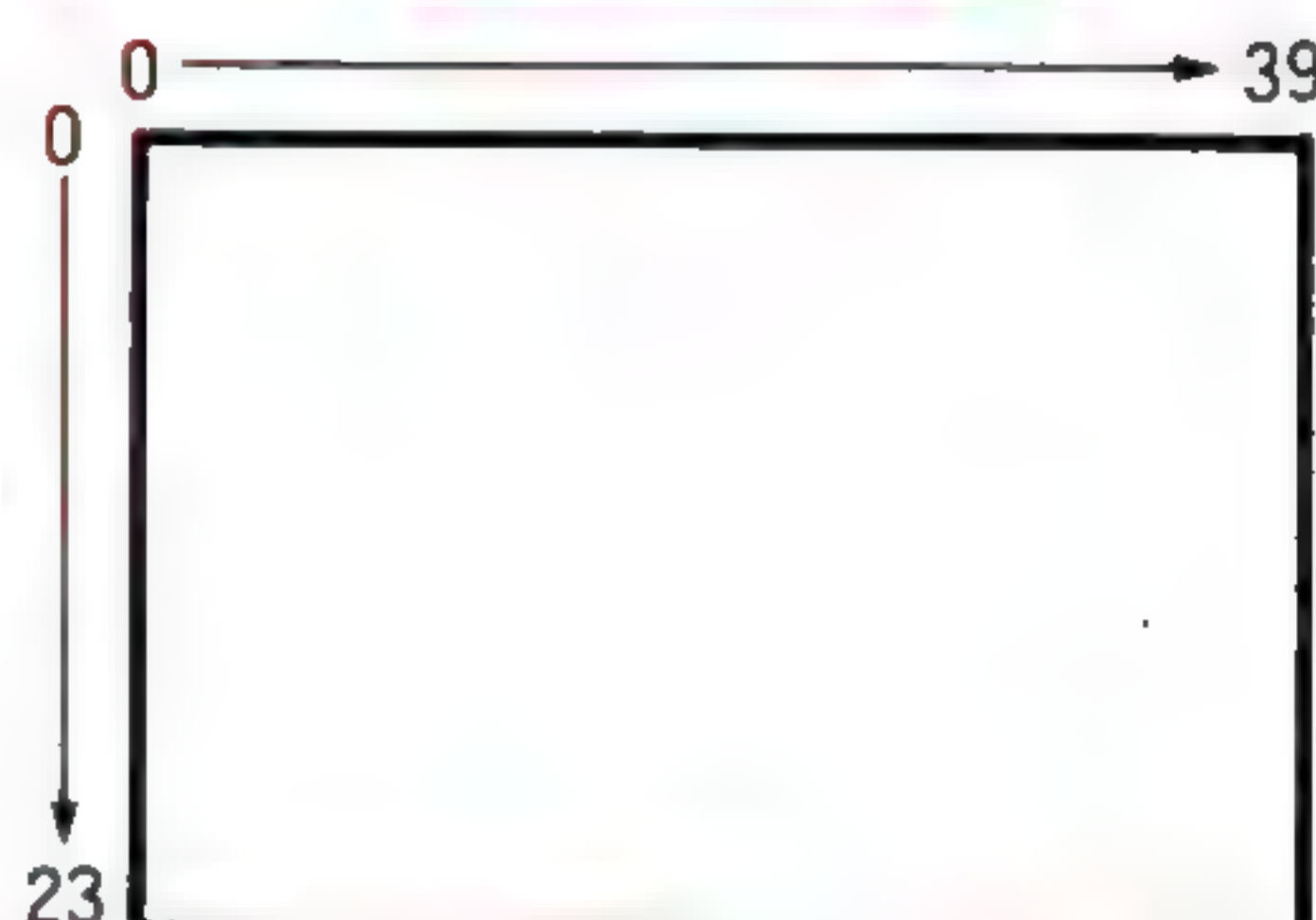
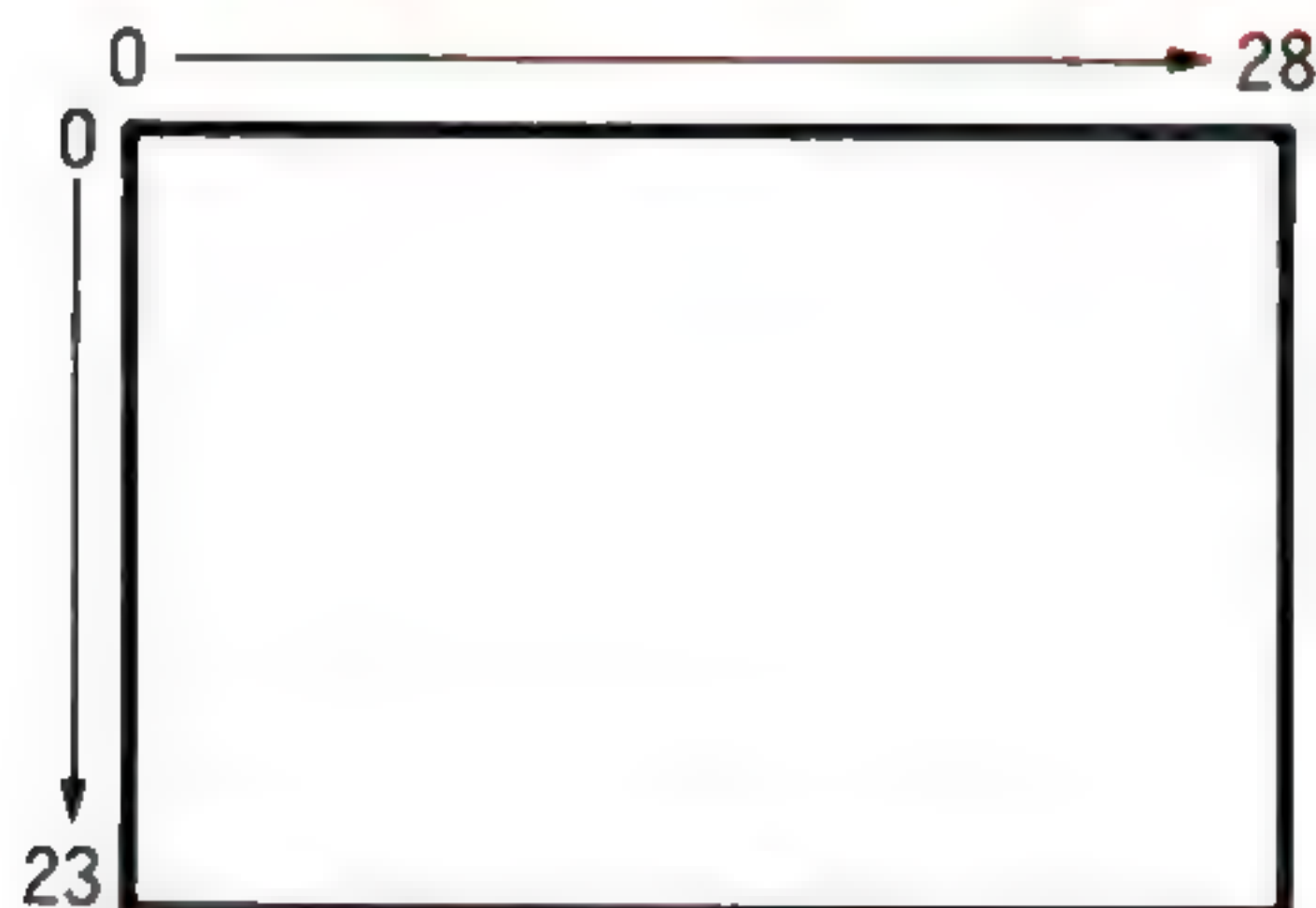
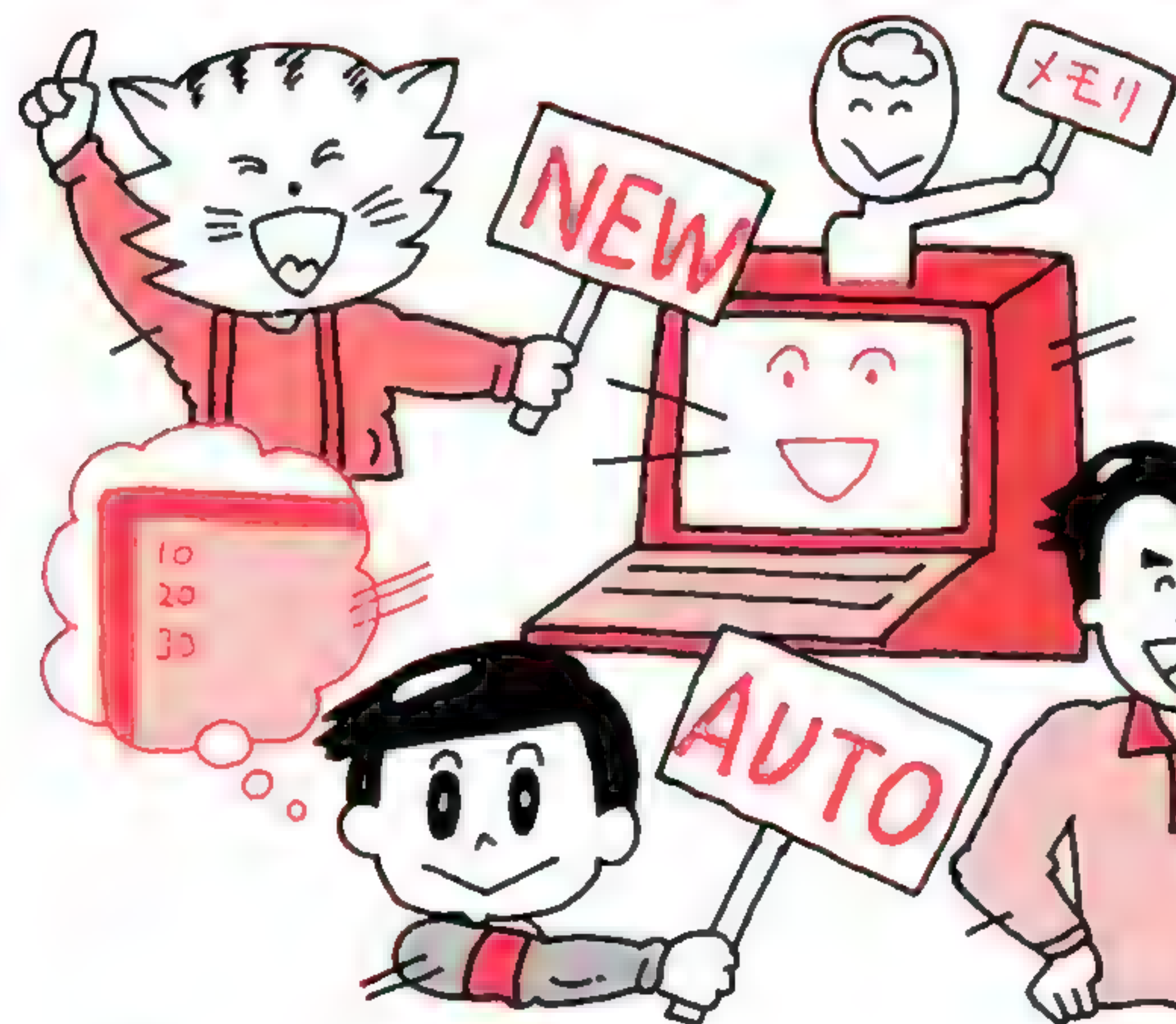
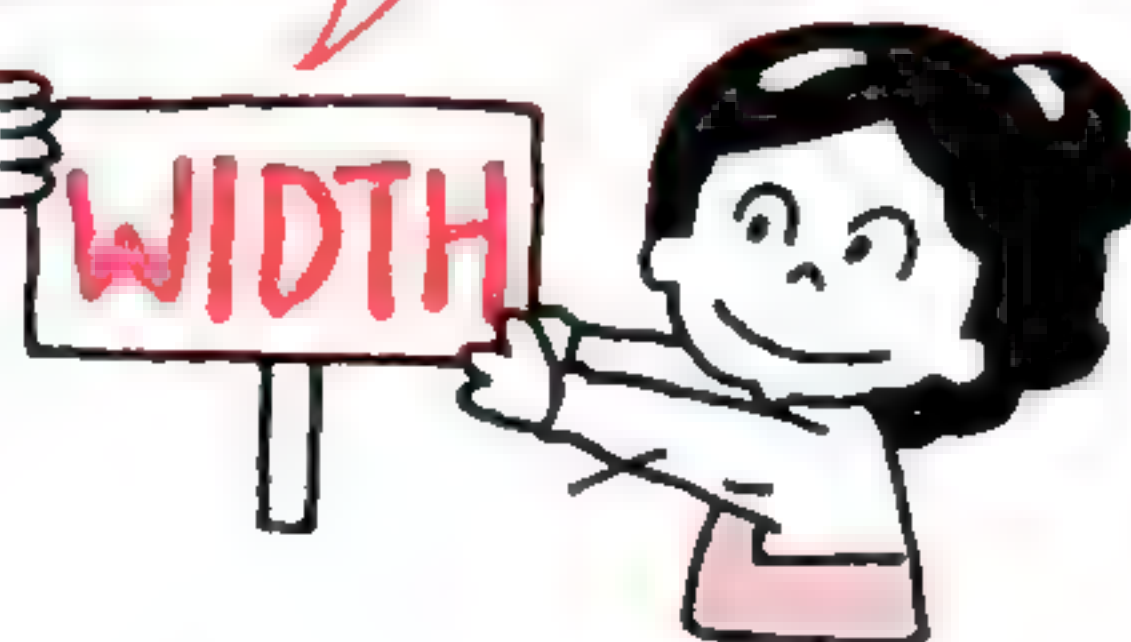
行番号を自動的につけてくれるのが、AUTO命令だ。

[A][U][T][O]とキー入力して、[RETURN]キーを押すと、行番号は自動的に10から始まって、そのあとは、[RETURN]キーを押すごとに、10、20、30、40………と10きざみに行番号が付けられる。

そのほか、出だしの行番号やきざみの数を指定すれば、いろいろな行番号のつけ方もできるよ。

AUTO命令を取り消したいときは、[CTRL] キーと [STOP] キーを同時に押すんだよ。

WIDTH命令●画面に表示させる桁数を決める

WIDTH命令は、画面に表示させる桁数を決めるんだ。つまり、横に表示できる文字数ってことだね。

画面だってワイドになるんだ

WIDTH　40　を実行すると、横に40文字分、入力できるようになるよ

パート2

BASICを マスターしよう

とびとびにプログラムを実行

GOTO 50 なら50行に行く

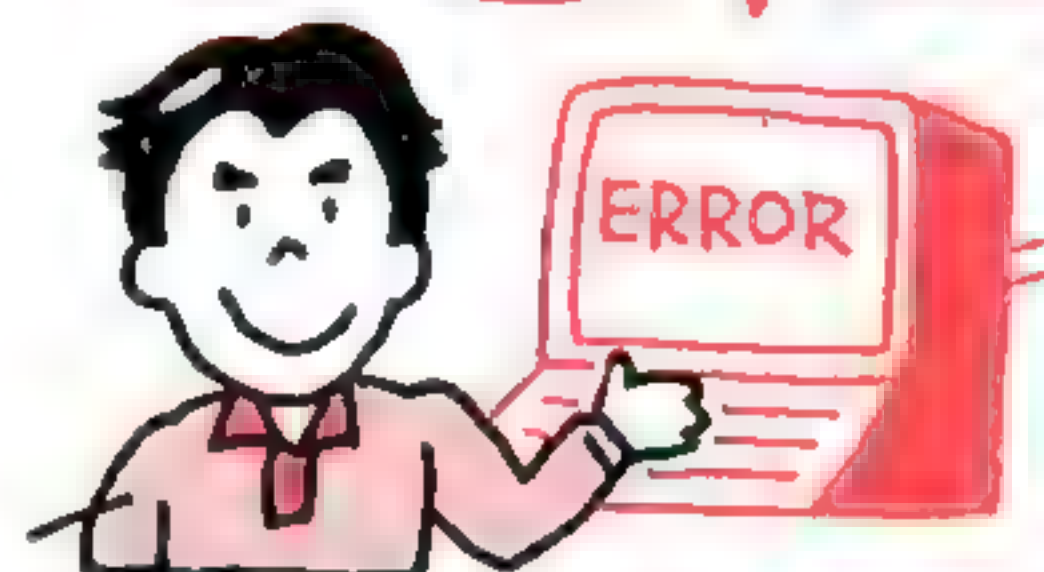
GOTO命令●指定された行にジャンプする

プログラムの行番号は知っているよね。プログラムの各行の行頭にかならずついていて、最初は小さな数字で、後ろになるにしたがって、だんだん大きな数字になっていく。

そして、パソコンは、行番号の小さなものから順に実行していく——ということになっている。でも、順序を気にかけずに、好きな行番号のところに行ってしまう場合がある。

たとえば、GOTO命令だ。GOTO命令はジャンプの名選手だ。

```
10  GOTO  100
```

とすれば、10行の後ろに20行があっても、30行があっても、そこには見向きもしないで、GOTOの次に書かれた100という行にジャンプして行ってしまう。つまり、プログラムの行番号の順序は考えずに、指定した行に飛んで、その行を実行するというわけだ。英語でGO TOは、〜へ行けということだから、同じようなものだね。

●GOTO文ショートプログラム

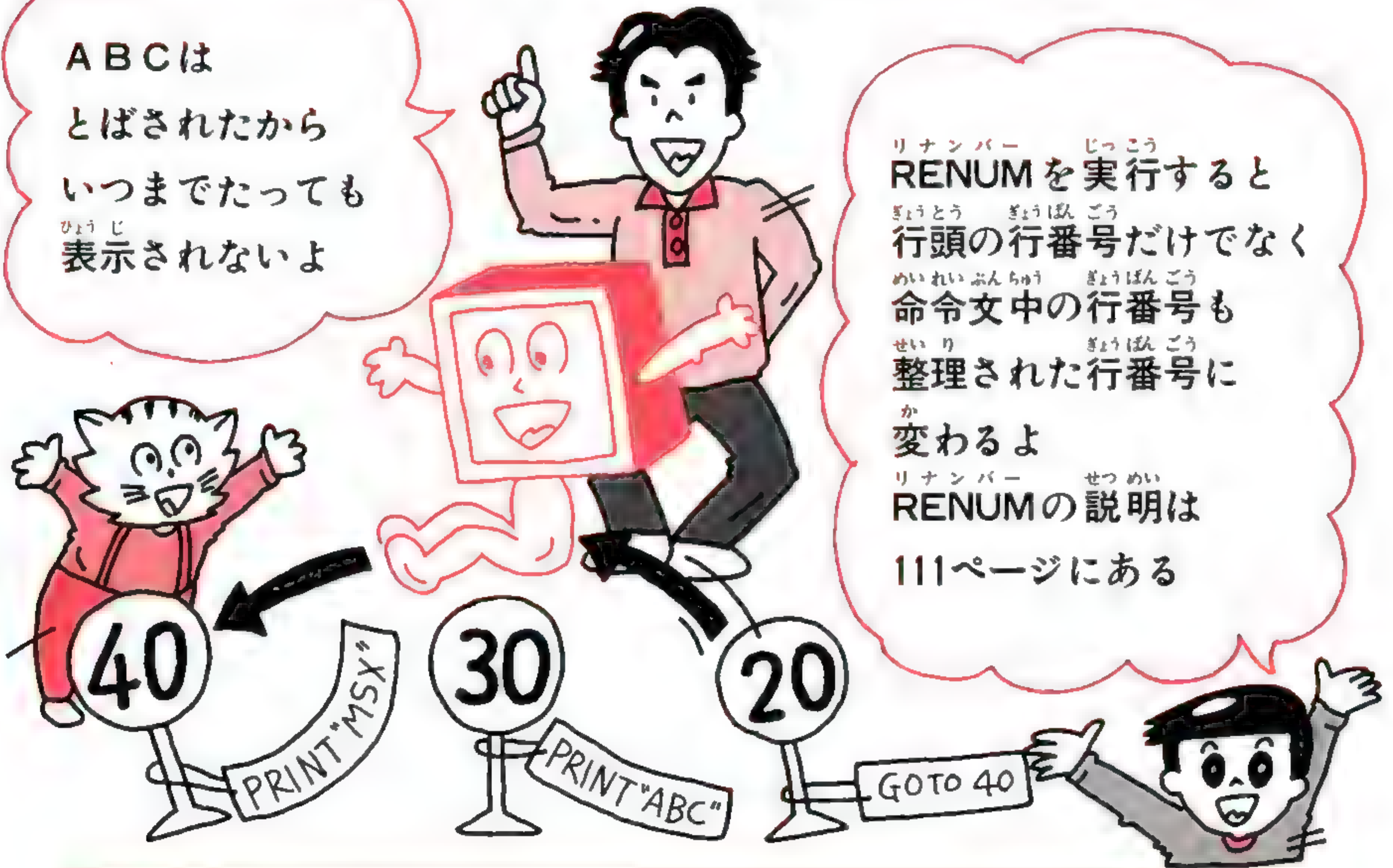
```
10 CLS
20 GOTO 40
30 PRINT "ABC"
40 PRINT"MSX"
50 END
```

左のプログラムは、20行のGOTO文で、30行を飛び越えて、40行にジャンプし、40　PRINT〝MSX″を実行する。

```
MSX
```

だから、画面には〝MSX″とだけ表示するが、30行のPRINT〝ABC″は飛び越されたので、実行されない。

ABCは
とばされたから
いつまでたっても
表示されないよ

RENUMを実行すると
行頭の行番号だけでなく
命令文中の行番号も
整理された行番号に
変わるよ
RENUMの説明は
111ページにある

GOTO命令

GOTO文につづいて指定した行番号の行にジャンプして、その行の命令を実行する。

GOTO文は、GOとTOの間を1字だけあけて、GO　TOと書いてもいいが、2字以上あけて書くと、GOTO文にはならない。

ジャンプしてふたたび戻ってくる

GOSUB～RETURNとおぼえる

GOSUB命令●RETURNで戻って次の命令から

どこへでも、指定する行へジャンプするのが、GOTO命令だったね。ただ、GOTO命令はジャンプして行ったら、行きっぱなし。もとに戻ってはこない。

GOSUB命令は、GOTO命令と同じで、指定の行にジャンプするが、その行からの命令を実行したあと、RETURN文があれば、いちどもとのところに帰ってきて、GOSUB命令のあった次の命令から実行していくのだ。だから、GOSUB命令はGOSUB～RETURNと、ひとまとめにして、おぼえておいたほうがいい。

たとえば、

10　GOSUB　100

として、100行へジャンプして、100行から実行したあと、

150　RETURN

と、RETURN命令があったら、もとに引き返して、GOSUB命令の次の20行から実行するわけだ。

GOSUBは
サブルーチンへ行け
RETURNは
戻れということだよ

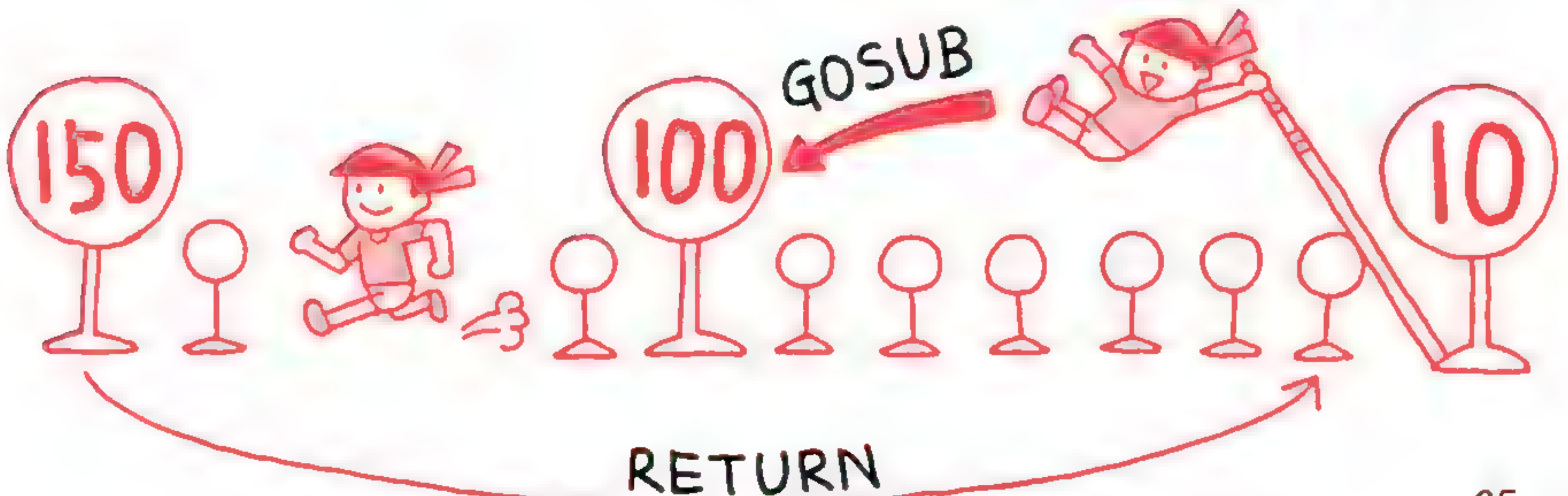
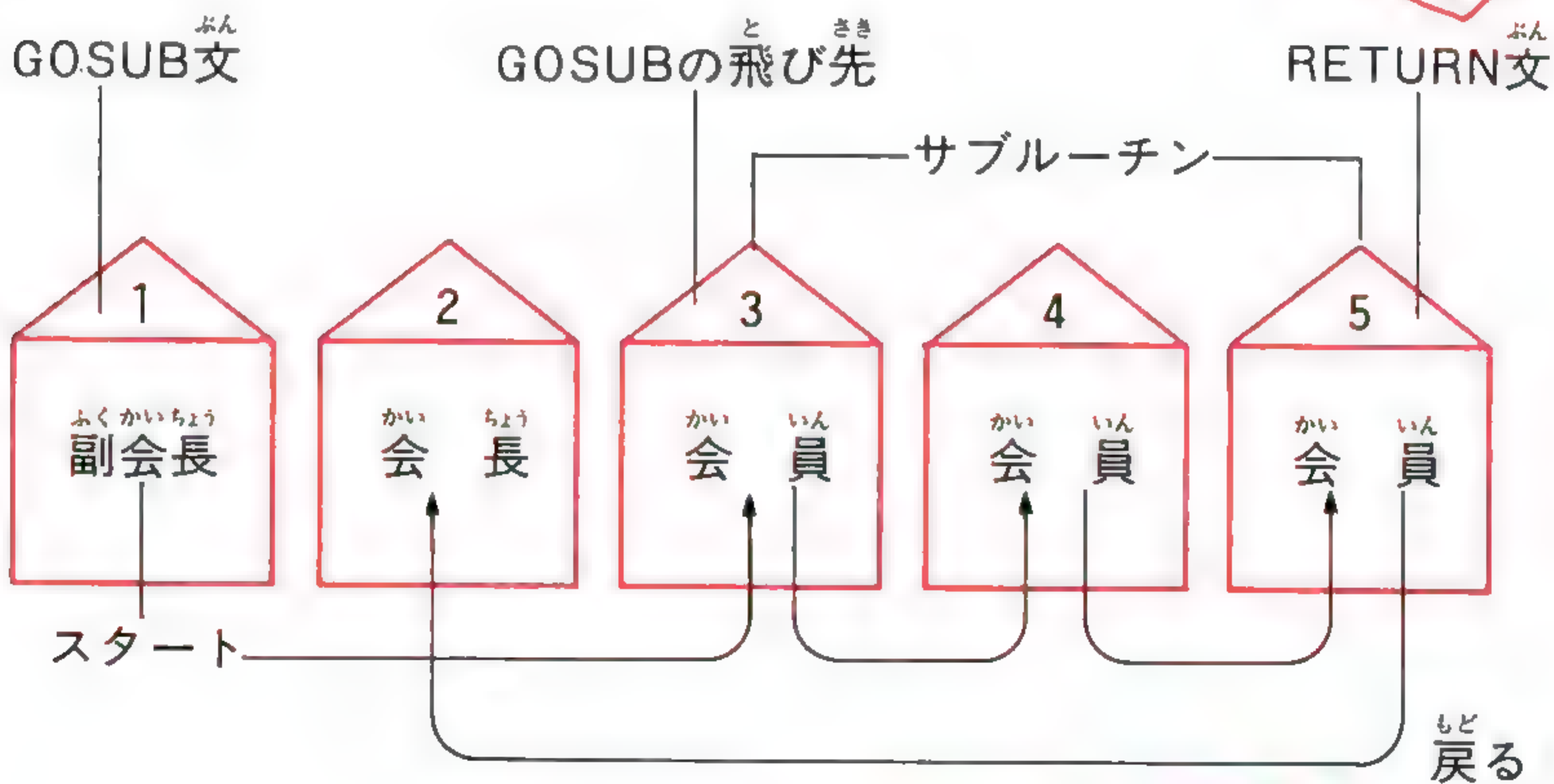
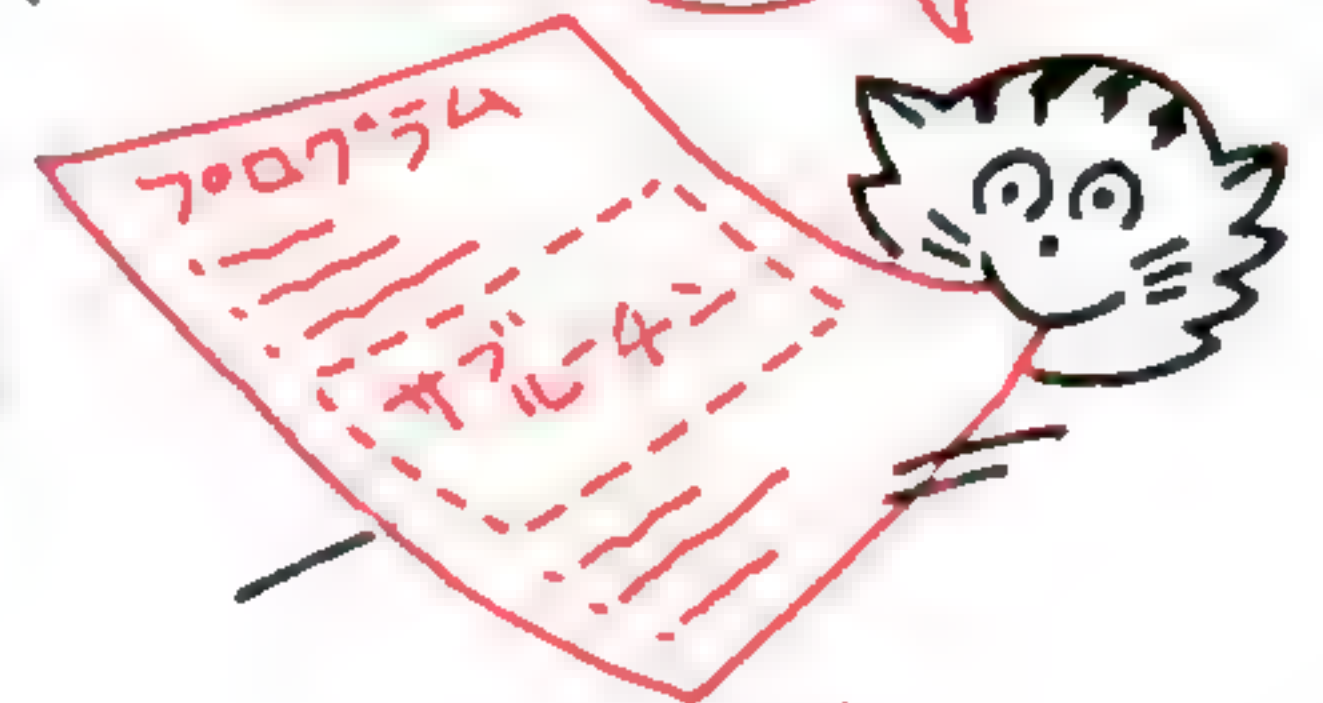
GOSUB　100は
100行に
ジャンプしろだ

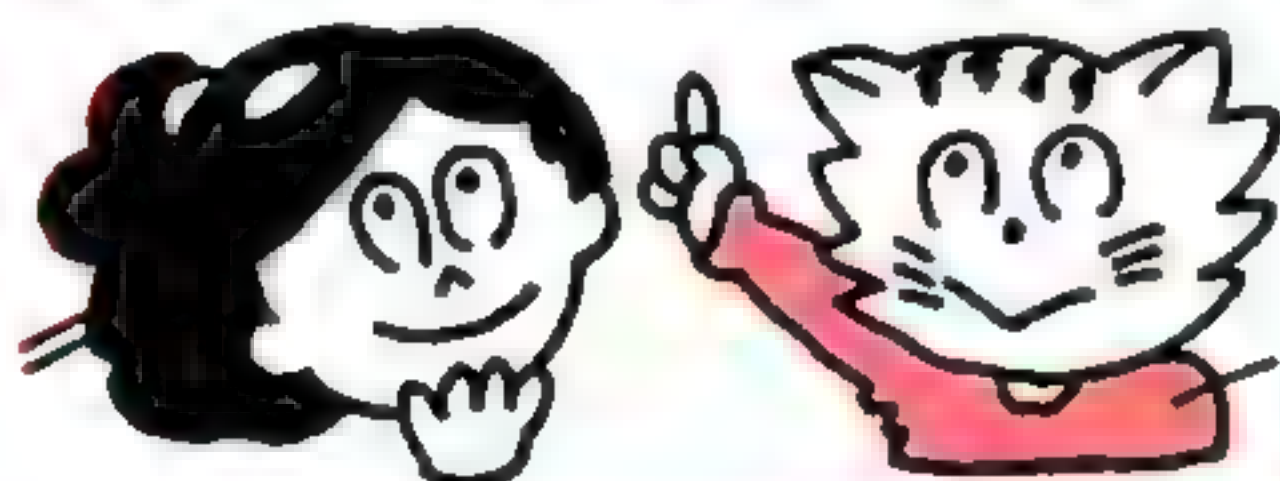
会長の家は会員をまわったあとで

左から1、2、3、4、5と5軒並んだ家があったとしよう。1は町内会の副会長さん、2は会長さん、3、4、5は会員さんだ。

副会長さんが作った文書に全員の名まえを書いてもらい、最後に会長さんの名まえとハンコをもらうとき、どうする？

1から出発して、2は会長さんだからとばして、3、4、5とまわったあと、2の会長さんのところに戻って、名まえとハンコをもらう、という順序だ。

サブルーチンって、それだけで独立したプログラムの中の小さなプログラムだよ

●GOSUB文ショートプログラム

下のプログラムは、＊ジルシで四角の枠を描き、その中に〝ワタシハ〟〝ＭＳＸ〟〝ヨロシク〟と、順番に書いたり、消したりしている。

```
10 CLS
20 GOSUB 110
30 LOCATE 4,3:PRINT"ワタシ ハ"
40 GOSUB 190:CLS
50 GOSUB 110
60 LOCATE 4,3:PRINT"M S X"
70 GOSUB 190:CLS
80 GOSUB 110
90 LOCATE 5,3:PRINT"ヨロシク"
100 GOSUB 190:END
110 PRINT"*************"
120 PRINT"*           *"
130 PRINT"*           *"
140 PRINT"*           *"
150 PRINT"*           *"
160 PRINT"*           *"
170 PRINT"*************"
180 RETURN
190 FOR N=1 TO 1000:NEXT N
200 RETURN
```

①四角の枠を表示するサブルーチン

②カウンターのサブルーチン

①のサブルーチンと②のサブルーチンが何度も呼び出される

10→20→[110～180]→30→[40 GOSUB 190]→[190～200]→[40 CLS]

→50→[110～180]→60→[70 GOSUB 190]→[190～200]→[70 CLS]……

マルチステートメントに注意。GOSUBの次の命令に戻るんだ。

サブルーチンの使い方

サブルーチンを呼び出すのがGOSUB文だとわりきったほうが、わかりやすい。GO（行け）SUB（サブルーチンへ）だから当然そうなのだ。

サブルーチンは、1個の独立したプログラムだとは、まえにも書いたね。だから、プログラムの中で、何度も使うような内容があったら、サブルーチンにしておけば、使いたいとき、いつでも、GOSUB文で呼び出すことができて、たいへん便利だね。

それから、1つのサブルーチンの中に、もうひとつサブルーチンを作って、これを呼び出すこともできる。これを、サブルーチンの多重化といって、ちょっと高度なテクニックだよ。

サブルーチンの中にCLS文を入れたらいけないよ

サブルーチンをGOTOやRUNで実行すればエラーだ

GOSUB命令

GOSUBのあとに飛び先の行番号を指定する。

GOSUB　50

50行に飛んで、50行以降のサブルーチンを実行したあと、RETURN文があったら、GOSUB文の次の命令に戻る。

1つのサブルーチンの中には、複数のRETURN文があってもいい。RETURN文では、行番号を指定して、特定の行番号にジャンプさせることもできる。

あきずに同じ動作をくり返せ

たとえば戦車を右に動かしたり

FOR〜NEXT●くり返す動作の回数を指定して

FOR〜NEXTは、FORとNEXTにはさまれた命令を、指定した回数だけくり返して実行させる命令だ。

FOR　N＝1　TO　100

として、次にくり返して実行させる命令を書き、

NEXT　N

とすれば、FORとNEXTの間の命令を、N回だけくり返す。そして、Nは1から100までだから、Nが1のときに実行、Nが2のとき、3のとき………Nが100のときと、100回実行するということになる。

Nは変数だよ
変数だから
状況しだいで
変化する数だ

たとえば、

```
[#]--
[#]
<*****>
```

と、こんな戦車があったとして、これを画面に表示して動かすなどというときに使える命令だ。

FOR〜NEXTの
あいだに
戦車を描いて消す
動作を入れればいい

動かすというのは、表示を消して、その隣に同じ表示を描いて、また消して、隣に描いて、消して……という動作をくり返せばいいのだから——。

●FOR～NEXT文ショートプログラム

```
10 CLS
20 FOR N=1 TO 20
30 LOCATE N,10
40 PRINT"  [#]--"
50 LOCATE N,11
60 PRINT"  [#]"
70 LOCATE N,12
80 PRINT"<*****>"
90 FOR I=1 TO 200:NEXT
100 LOCATE N,12
110 PRINT" "
120 NEXT
130 LOCATE N-1,12
140 PRINT"<"
```

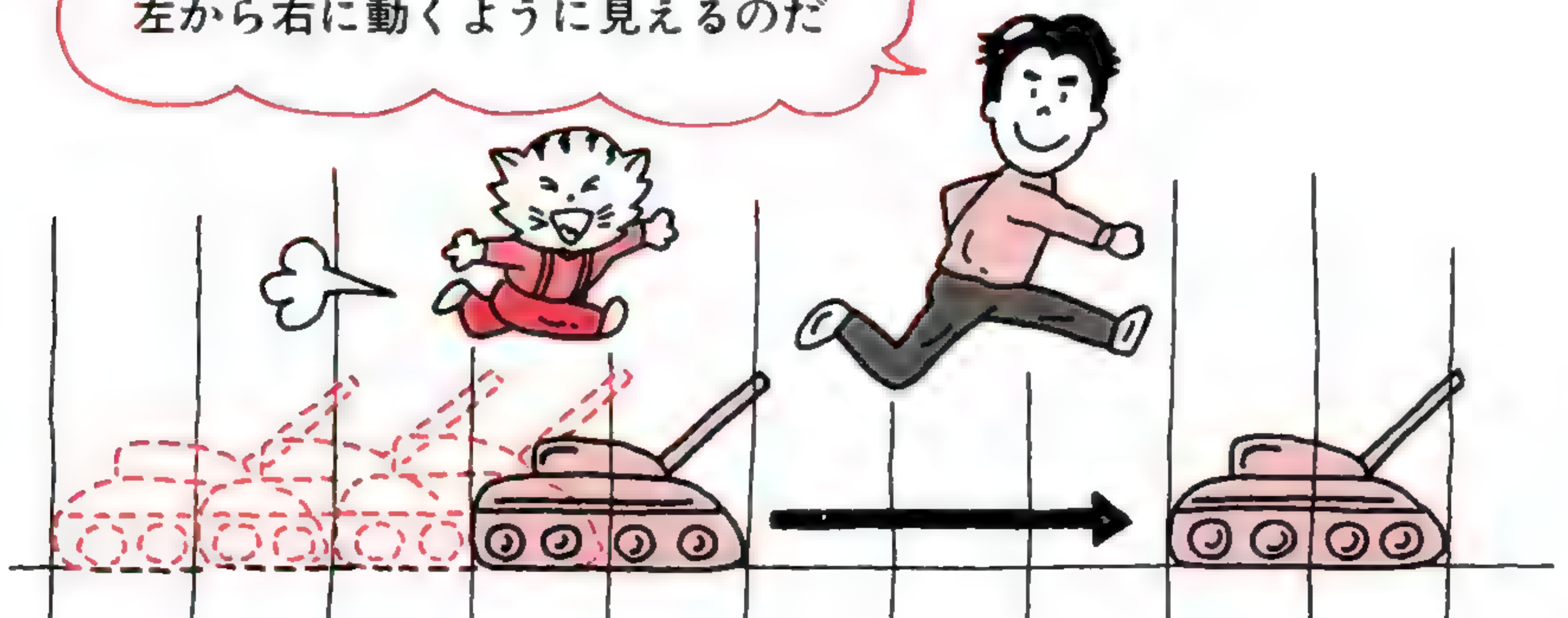
FOR～NEXT文の中に、もう1つのFOR～NEXT文が90行にある。

90行のFOR～NEXT文は、この行内を1から200まで行ったり来たりして、時間かせぎをしているループカウンターだ。

20行のFOR文と120行のNEXT文のあいだで、

```
 [#]--
 [#]
<*****>
```

という戦車みたいなキャラクタを描いて消し、次に1だけ右の座標に同じものを描いて消して、これを20回くり返している。

戦車みたいなキャラクタが右に動く
描いて消して、右に1ずれた座標に
描いて消すから
左から右に動くように見えるのだ

FOR～NEXTの入れ子構造

●FOR～NEXTの中にまたFOR～NEXT

FOR～NEXTの中に、いくつでも、FOR～NEXT文を入れて使うことだってできる。これを入れ子構造っていうんだ。

例をいくつか、あげておこう

```
(1) 10  FOR  I=0  TO  100
    20  FOR  J=1  TO   50
     :
    60  NEXT  J
    70  NEXT  I
(2) 10  FOR  I=0  TO  100
    20  FOR  J=1  TO   50
     :
    60  NEXT  J, I
(3) 10  FOR  I=0  TO  100
    20  FOR  J=1  TO   50
     :
    60  NEXT
    70  NEXT
```

中に入れるFOR～NEXTは完全に外側のFOR～NEXTの中に入っていなければだめだよ

中と外のFOR～NEXTが同じところで終わるならNEXT文を1つにまとめてもいい
変数の順序は中の変数、外の変数の順だ

外と中の変数はかならずちがった変数を使うこと

NEXT文の変数は省略できるよ
このときのNEXTはいちばん近くにあるFORのものだ

●FOR〜NEXTのループ回数に注意する

FOR　J=10　TO　1………初期値が最終値より大きいときは、1回だけ実行される。

FOR　J=−10　TO　−5………小さい値から大きい値までの数、つまり−10から−5までの数5回のループをする。

FOR　J=10　TO　10………初期値と最終値が同じで増分0のときは、1回だけ実行される。

FOR〜NEXT文を使うときは変数の初期値と最終値に注意しよう

FOR〜NEXT命令

FOR〜NEXT命令の間の命令を、変数で指定した回数だけくり返す。FORのあとの変数はIでもNでもよいが、NEXTの変数と同じにする。

FOR　変数=初期値　TO　最終値　STEP　増分
〜
NEXT　変数

初期値の数字から最終値の数字までが、くり返す回数。STEP 増分は、初期値の数から最終値の数までのステップを示す。たとえば、0　TO　10　STEP　2なら、変数0に2プラスして変数2、2プラスして変数4………変数10と実行する。STEPは省略できる。STEPを省略したときは、増分は1とみなされる。

天気だったら野球 雨だったら勉強だ

条件しだいで行き先を決める

IF～THEN●IF～THEN～ELSE

もし、キミが100点とったら、ボクはアンパンを100個たべるとか、あした天気がよかったら、サッカーをしようとか、そのときの状況しだいで、あれをやったり、これをしたりするのが、IF文だ。

ＩＦ文には、大きく分けて、

ＩＦ～ＴＨＥＮ

ＩＦ～ＴＨＥＮ～ＥＬＳＥ

と２とおりの文型がある。

もし（IF）天気だったら、そのときは（THEN）野球をしよう、というのがIF～THEN。

もし、あした天気だったら、野球をしよう、でも雨だったら（ELSE）勉強しよう、というのが、IF～THEN～ELSEなのだ。

ＩＦ　Ａ＝５　ＴＨＥＮ　100

というふうに使う。変数Ａが５のときは、プログラムの100行に行けという意味だ。条件を判断して、それからどうするかというわけだ。

この＝は、代入ではなく等しいの意味だよ。

IF～GOTO
という形もあるよ
「Aが５なら
100行に行け」は
ＩＦ　Ａ＝５
ＧＯＴＯ　100だ

●IF～THENの使い方

■2つの数値や文字列の大小を比較する。

```
IF  A<100  THEN  A=A+10
```

（Aが100より小さかったら、Aに10を加える）

```
IF  A$="YES" THEN  100
```

（変数A＄の内容が"YES"だったら、100行にジャンプする）

比較演算子の
＝や＜や＞を使って
比較するのだ

●数値が0かどうかを判断する

```
IF  A<>0  THEN 150  ELSE  200
```

（Aが0でなかったら150行へ、0だったら200行にジャンプする）

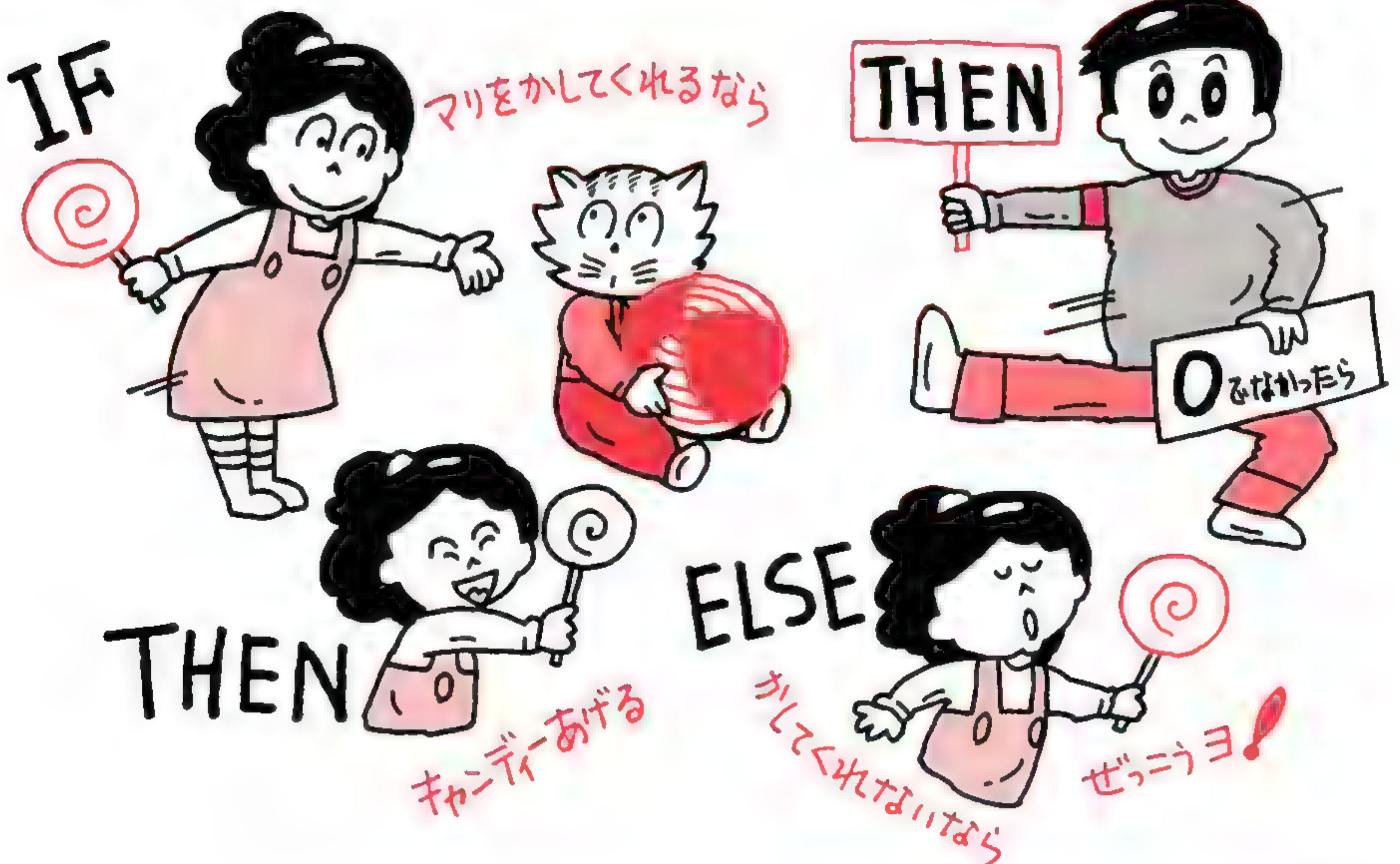
```
IF  A+B<>0  THEN  200
```

（変数A＋Bが0でなかったら200行にジャンプする）

IF　マリをかしてくれるなら

THEN　0でなかったら

THEN　キャンディーあげる

ELSE　かしてくれないなら　ぜっこうヨ！

●IF～THEN文ショートプログラム

```
10 CLS:PRINT"U,H,K,Mノ キーヲ オス"
20 K$=INKEY$:IF K$=""THEN 20
30 IF K$="u" THEN 70
40 IF K$="h" THEN 80
50 IF K$="k" THEN 90
60 IF K$="m" THEN 100 ELSE 20
70 PRINT"ウエ デ゛ス":GOTO20
80 PRINT"ヒダ゛リ デ゛ス":GOTO20
90 PRINT"ミギ゛ デ゛ス":GOTO20
100 PRINT"シタ デ゛ス":GOTO20
110 PRINT"シタ デ゛ス":GOTO20
```

INKEY$(インキーダラー)はキーボードで押(お)されたキーの最初(さいしょ)の１字(じ)をつかまえる関数(かんすう)だ

まず〝U、H、K、Mノキーヲオス〃と表示(ひょうじ)されるから、そのとうりに、まず「U」のキーを押(お)そうじゃないか。

```
U,H,K,Mノ キーヲ オス
```

```
ウエ デス
```

ほら[U]のキーを押(お)したよ
あれ、〝ウエ　デス〃って
いったいこれはなんだ？

IF～THEN～ELSE命令(イフ ゼン エルス めいれい)

与(あた)えられた条件(じょうけん)がどうなのかを判断(はんだん)して、次(つぎ)の行(い)き先(さき)を決(き)める。

IF 条件(じょうけん) THEN 文(ぶん)または行番号(ぎょうばんごう) ELSE 文(ぶん)または行番号(ぎょうばんごう)

与(あた)えられた条件(じょうけん)に合(あ)っていたら、THENのあとの文(ぶん)または行番号(ぎょうばんごう)を実行(じっこう)し、そうでなかったら、ELSEの次(つぎ)を実行(じっこう)する。

「ウエデス」っていったいなんだろう。わかっている人は、スルドイぞ。

U、H、K、Mのキーボードの配置を見てほしい。わかったね。4文字のうち、[U]は、上にあるではないか。

[H]を押してみよう。表示は、〝ヒダリデス〟だね。[K]を押せば〝ミギデス〟[M]を押せば〝シタデス〟だ。

U

H　　K

M

4個のキーのうち、Uのキーは上にある。Hが左で、Kは右、Mは下なのだ

プログラムを見てみよう。

20行　キーが押されなかったら、プログラムがストップして、永遠のキー入力待ちだ。

30行　[U]のキーが押されたら、（変数K＄が〝u〟なら）、70行にジャンプ。70行で〝ウエデス〟と表示して、また20行に戻って、キー入力待ちだ。

40行　[K]が押されたら……、[U]と同じだ。80行で〝ヒダリデス〟と表示して、20行に戻る。あとは、これと同じくり返しだ。

点でつづる自由なお絵描き

●IF~THEN文のお遊びプログラム

```
10 CLS:KEY OFF
20 SCREEN 0
30 WIDTH 40
40 COLOR 15,5
50 X=10:Y=10
60 LOCATE X,Y
70 PRINT"*";
80 ST=STICK(0)
90 A=0:B=0
100 IF ST=1 THEN A=0:B=-1
110 IF ST=5 THEN A=0:B=1
120 IF ST=7 THEN A=-1:B=0
130 IF ST=3 THEN A=1:B=0
140 LOCATE X,Y
150 PRINT".";
160 IF X+A<0 OR X+A>39 THEN A=0
170 IF Y+B<0 OR Y+B>22 THEN B=0
180 X=X+A:Y=Y+B
190 LOCATE X,Y
200 PRINT"*";
210 GOTO 80
```

RUNさせると、画面に*ジルシ（アスタリスク）が表示されるから、これを画面いっぱい前後左右にとびはねさせるのだ。*を動かすキーはカーソルキー。

*が動きまわったあとには、「・」が残るから、どんなおもしろい図でも、おもしろくない図でも、きみの腕しだいで描ける。

100行からのIF文にあるST=1、ST=5、ST=7、ST=3は、*の進む方向、上下左右を条件にしている。

ひと筆描きのように描けばいいのだな

並んだデータを順番どおりに呼び出す

変数の順とデータの順は一致する

READ～DATA●2つそろっていちにんまえ

READ文は、DATA文に書かれたたくさんのデータを呼び出して、READ文にある変数に順序よく代入する命令だ。

READ文があったら、かならずDATAがあるのだから、READ～DATAと、ひとまとめにして、おぼえておこう。

次のプログラムを入力して、RUNさせよう。

```
10  READ  A
20  READ  B
30  PRINT  "A=" ;A
40  PRINT  "B=" ;B
50  DATA  8
60  DATA  20
```

```
A=8
B=20
OK
```

READ文の最初の変数はDATA文の最初のデータを呼び込むのだよ

データ8が変数Aに代入　データ20が変数Bに代入されていくんだって

10行のREAD文にある変数Aに、50行のDATA文のデータ8（定数）が代入され、20行のREAD文にある変数Bに、60行のデータ20が代入されたんだよ、ということを、画面の表示が示しているのさ。

筆が変数なら色がデータだ

●虹の色を塗る順番も決まっている

READ~DATAの使い方を、もう少しくだいて説明してみよう。

絵の具箱がある。いろいろな色の絵の具がある。

虹を描いてみようかな、とおもう。虹の7色は、上から、赤、オレンジ、黄、緑、水色、青、紫の順だ。

色がまじるときたなくなるから、7本の筆を用意して、1本の筆で1色の色を塗ろう。1の筆は赤用だ。2の筆はオレンジ用だ。3の筆は……。

READ文に書く変数が筆だ。

DATA文に書くデータが絵の具の色だ。

書かれた変数の順番に合ったデータが、順番どおりに、それぞれの変数に代入されていく。

READ A, B, C, D…

READ文の変数は
コンマ(,)で区切って
たくさん
書いてもいい

●READ〜DATA文ショートプログラム

棒グラフを描いてみよう

```
100 REM ﾎﾞｳ ｸﾞﾗﾌ
110 CLS
120 SCREEN 0,0:WIDTH 40
130 FOR N=5 TO17 STEP 2
140 H=H+1
150 READ A
160 LOCATE 3,N:PRINT H;":";
170 FOR I=1 TO A
180 PRINT "*";
190 NEXT I
200 NEXT N
210 DATA 15,8,19,14
220 DATA 9,26,16
```

FOR〜NEXTは
130〜200行
ここで7回まわって
データを7個
ひろってくる

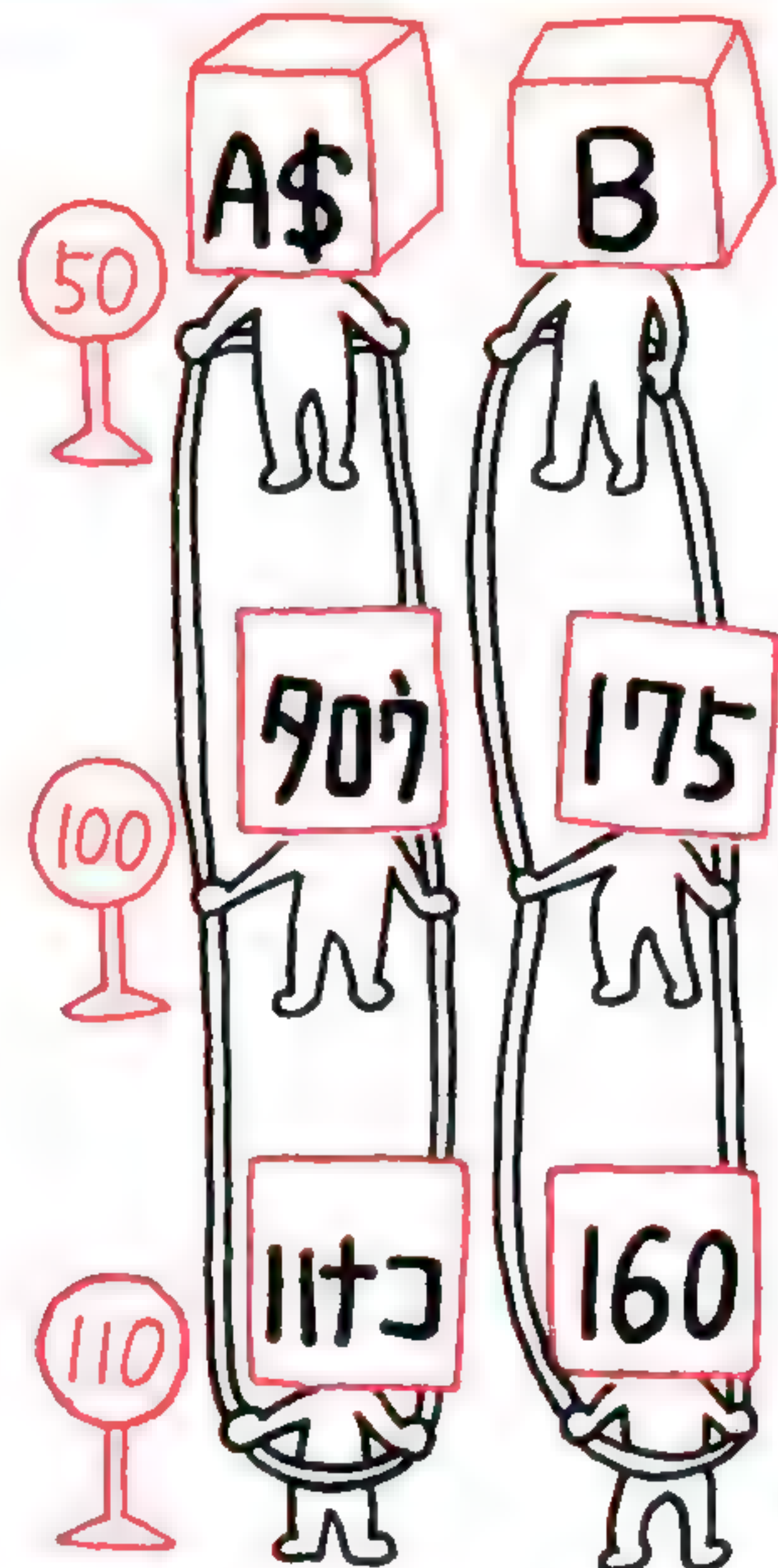
READ文の変数Aに
210、220行の
データが7個
順番に入ってくる

```
1 :＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊
2 :＊＊＊＊＊＊＊＊
3 :＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊
4 :＊＊＊＊＊＊＊＊＊＊＊＊＊＊
5 :＊＊＊＊＊＊＊＊＊
6 :＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊
7 :＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊
```

160 LOCATE 3,N:PRINT H;":";

140　H＝H＋1
7項目の数が1から1ずつふえて
160行のPRINT文で
書かれていくよ

座標（3，N）の位置から
1行とばしに
「：」ジルシを
表示せよ、だ

●READ～DATAのかたち

A＄は文字型変数、B、Cは数値変数だ。文字データを呼ぶのは文字変数、数値データを呼ぶのは数値変数なのだ

READ～DATA命令

READ文は、かならずDATA文がついてくる。

READ　変数, 変数, 変数………
～
DATA　データ, データ, データ………

という文型だが、このDATA文のデータが、READ文の変数に、順番どおりに代入される。READ文で使う変数は、数値変数でも文字変数でもいいが、読み込むデータと同じ型でなければ Syntax error となる。

READ～DATAを応用しよう

●電話管理プログラム

READ～DATAを使った応用プログラムを組んでみた。電話管理プログラムだ。

プログラムを短くするために、画面の大きさをWIDTH　20　と小さくしているが、ほぼ、実用に近いプログラムだ。

データをふやして
拡張すれば
電話帳代わりの
実用ソフトができる

```
10 CLS:SCREEN 0:WIDTH 20
20 DIM A$(5),B$(5):KEY OFF
30 FOR N=1 TO 5
40 READ A$(N),B$(N)
50 NEXT
60 CLS:PRINT" #### TEL ｶﾝﾘ ####"
70 LOCATE 0,5
80 PRINT"********************"
90 FOR I=1 TO 5
100 PRINT"*                  *"
110 NEXT
120 PRINT"********************"
130 LOCATE 0,15
140 INPUT A$:FOR N=1 TO 5
150 IF A$=A$(N) THEN 180
160 NEXT:LOCATE 0,15
170 PRINT SPC(20):GOTO 130
180 LOCATE 4,7:PRINT A$(N)
190 LOCATE 4,9:PRINT B$(N)
200 LOCATE 2,15:PRINT"ｷｰ ｦ ｵｼﾃ ｸﾀﾞｻｲ"
210 S$=INKEY$:IF S$="" THEN 210
220 GOTO 60
230 DATA ｺｸﾞﾏ,303-8909,ﾉｼﾞﾏ,876-9500
240 DATA ｼﾝｾｲ,831-0743,ｾｷｸﾞﾁ,800-1234
250 DATA ﾔﾏｼﾀ,42-0820
```

プログラムの使い方

プログラムをキー入力して、RUNさせたら、電話番号を調べたい人の名まえを入力する。その人の番号が、この電話帳に掲載されていれば、名まえと電話番号が表示されるよ。

もちろん、電話番号が登録されていない人の名まえを入力しても、それは表示されない。

電話番号を検索して表示したあと、「キーヲオシテクダサイ」と表示されるから、もう1度してみたかったら、どのキーでもいいから押せば、またスタートする。

このまま
調べられるのは
コグマとノジマ
シンセイにセキグチ
ヤマシタだけだよ

170行は
空白を20個
書いているんだよ
空白を20個書くから
表示された字が
消えてしまう

プログラムのかんたんな説明

・データの呼び込み 30〜50行のREAD文で、230〜250の行のデータを呼んでいる。

・名まえがあるかどうか 入力された名まえのデータがあるかどうかを判断するのは140〜160行のFOR〜NEXT文だ。名まえがあれば、180行に行って、名まえと電話番号を表示する。

ヤマシタ　42ー0820

・データをふやすなら 新しくデータ文で名まえと電話番号を追加し、20行のA＄（5）、B＄（5）、30行の5、140行の5とある数字の5をデータの数に合わせてふやせばいい。

まちがいの意味と行番号を表示

よくあるのはSyntax　error

エラーメッセージ●プログラムを直せばいい

人はまちがいをおかしたら、つぐないをしなければならない。でも、まちがったかどうかを知らせてくれなければわからない。この「オーイ！まちがっているよーん」と教えてくれるのが、エラーメッセージだ。

せっかく作ったプログラムを入力して、RUNさせはしたのだが……、画面には無情にもSyntax errorの表示。このエラーは、初心者ほどおかすエラーだ。

```
Syntax　error　in　10
```

Syntax errorは、入力したプログラムが、まちがって書かれていたというエラーだ。初心者のうちは、だれだってやることだから、はずかしがらなくたっていい。

[L][I][S][T][RETURN]として、プログラムを表示させ、まちがいを直せばOKだ。

よく出るエラーメッセージ

Syntax error

プログラムが文法に合ってないというエラー。実は、入力するときの打ちまちがいが多い。

Illegal function call

ステートメントや関数の使い方がまちがっている。指定できる範囲をこえて、指定してしまったなどがある。

Out of DATA

READ～DATAで、DATA文に読み出せるデータがない。

RETURN without GOSUB

GOSUB文とRETURN文が、うまくかみ合っていない。

Type mismatch

READ～DATAなどで、データの形式と変数の形式がまちがっている。

Device I/O error

カセットにSAVEしたプログラムが、うまく本体にLOADできない。本体とカセットの接続ミスや音量ミスのことが多い。

NEXT without FOR

NEXT文に対応するFOR文がない。

Undefined line numder

指定した行番号がプログラムにない。

パート3

やさしいプログラミング

プログラミングとは

命令をつないで仕事をさせること

フローチャート●仕事の流れを図に書いてみるとよくわかる

BASICの基本命令をひと通りマスターしたら、次はプログラムの作り方を勉強しよう。

プログラミングとは、コンピュータに目的の仕事をさせるために、命令と命令をつなぎ合わせてプログラムを作ることなのだ。

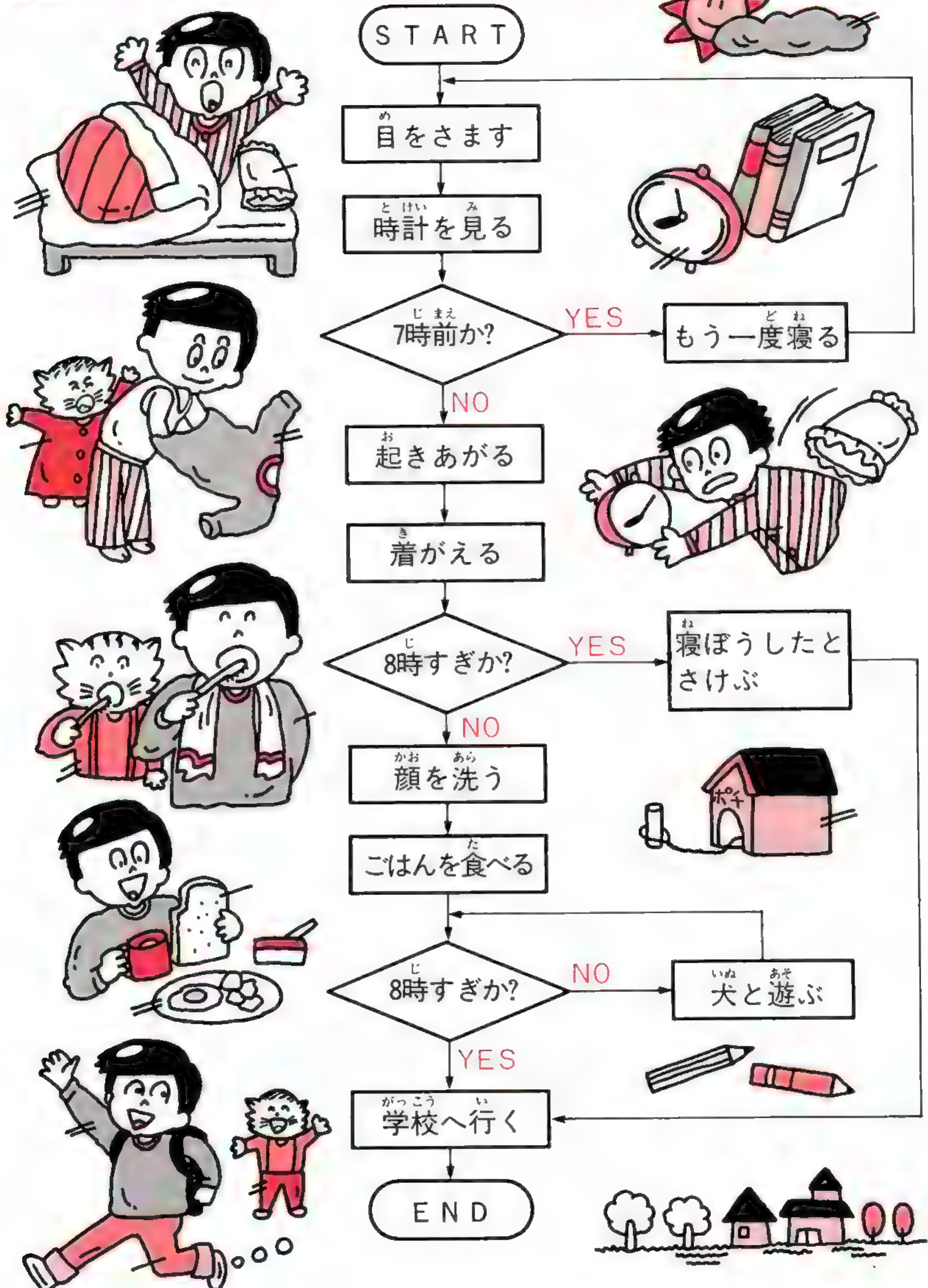
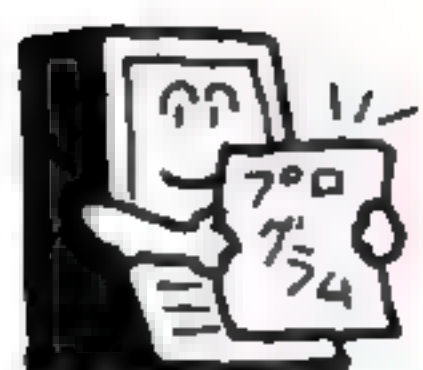
BASIC命令を知っていても、それをちゃんとつなぎ合わせることができなければ、プログラムを作ることはできないんだよ。

プログラムを作るには、まずどんな仕事をさせたいかをはっきり決める。次に、どんな順序でその仕事をさせるか、順番を決めてやる。その順番ごとに、BASIC命令をあてはめるんだ。

といっても、いきなりキーボードからプログラムを入力するのは、初心者のきみにはまだ早いのではないかな？

そこで、仕事の流れを図にあらわしてみよう。この図が「フローチャート」とか「流れ図」とよばれるものなんだ。

●フローチャートの書きかた

●朝起きて学校へ行くまでのフローチャート

三角形の面積を計算する

INPUT命令●キーボードからデータを入力する

INPUT命令は、キーを押してデータを入力するときに使う。

```
10 CLS                ①
20 INPUT A
30 INPUT B
40 PRINT A*B
```

```
10 CLS                ②
20 INPUT"A=";A
30 INPUT"B=";B
40 PRINT A*B
```

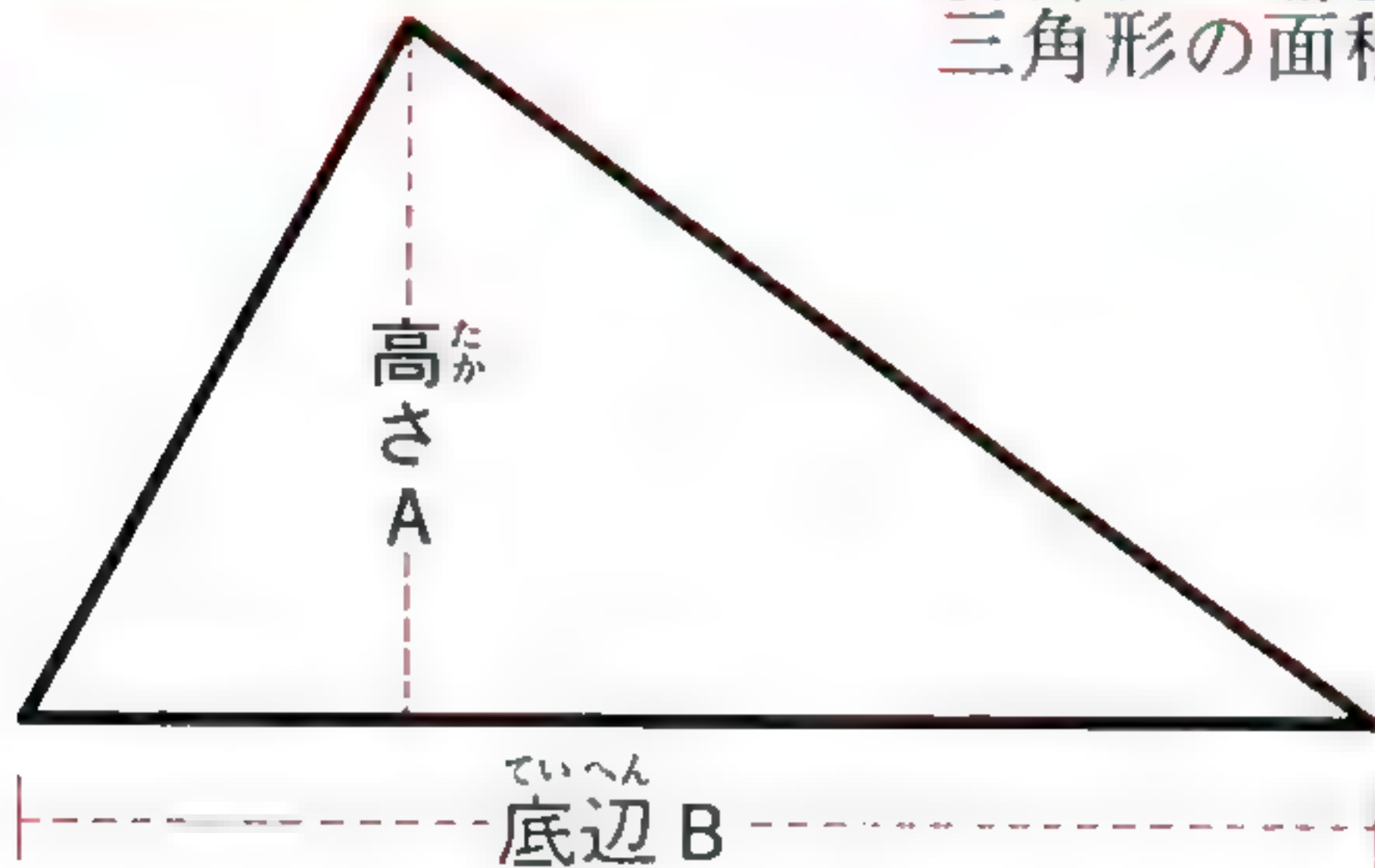
まず左のプログラムをRUNさせてごらん。画面は〝?〟（クェスションマーク）を出してとまったままだね。このままなにもしなければ、永遠に止まったままだ。

そこで、1でも2でもいいから、きみの好きな数字を入力してみよう。この数字が変数Aに代入される。

また?が出たね。もうひとつ数字をキー入力しよう。この数字が変数Bに代入される。

それから、40行の仕事、A×Bのかけ算をしてその答えを画面に表示してくれる。

右のプログラムをRUNさせてみよう。こんどは?だけのかわりに「A=?」、「B=?」と画面に表示される。つまりメッセージつきだから、意味がわかりやすいというわけだ。

AとBに入れたい数字をキーボードから入力すると、同じように計算して答えを出してくれるよ。

●三角形の面積を計算するフローチャート

三角形の面積をコンピュータに計算させよう。

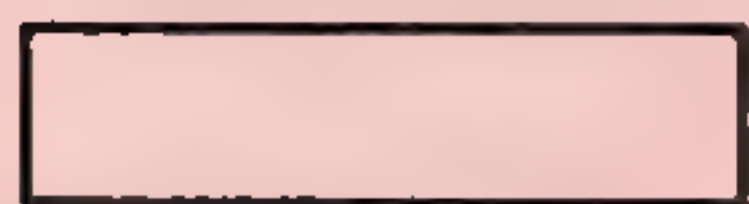
三角形の面積の計算は、「底辺×高さ÷２」だね。でも、このままをコンピュータにいったってわからないよ。

コンピュータに仕事をさせる順序でプログラミングしてやらなければならない。

コンピュータの仕事は、「キーボードから三角形の底辺の長さと高さを入力してやると、その面積を計算して画面に表示する」ということだな

フローチャートに使う記号

フローチャートを書くにも記号や規則があるけれど、いちばんたいせつなのは、ひと目見て流れがわかるということだ。あまり規則にしばられず、どんどん書いていこう。

(START)　(END)……………はじめと終わりの記号

［　　　］……………………………操作や仕事の内容を書き込む

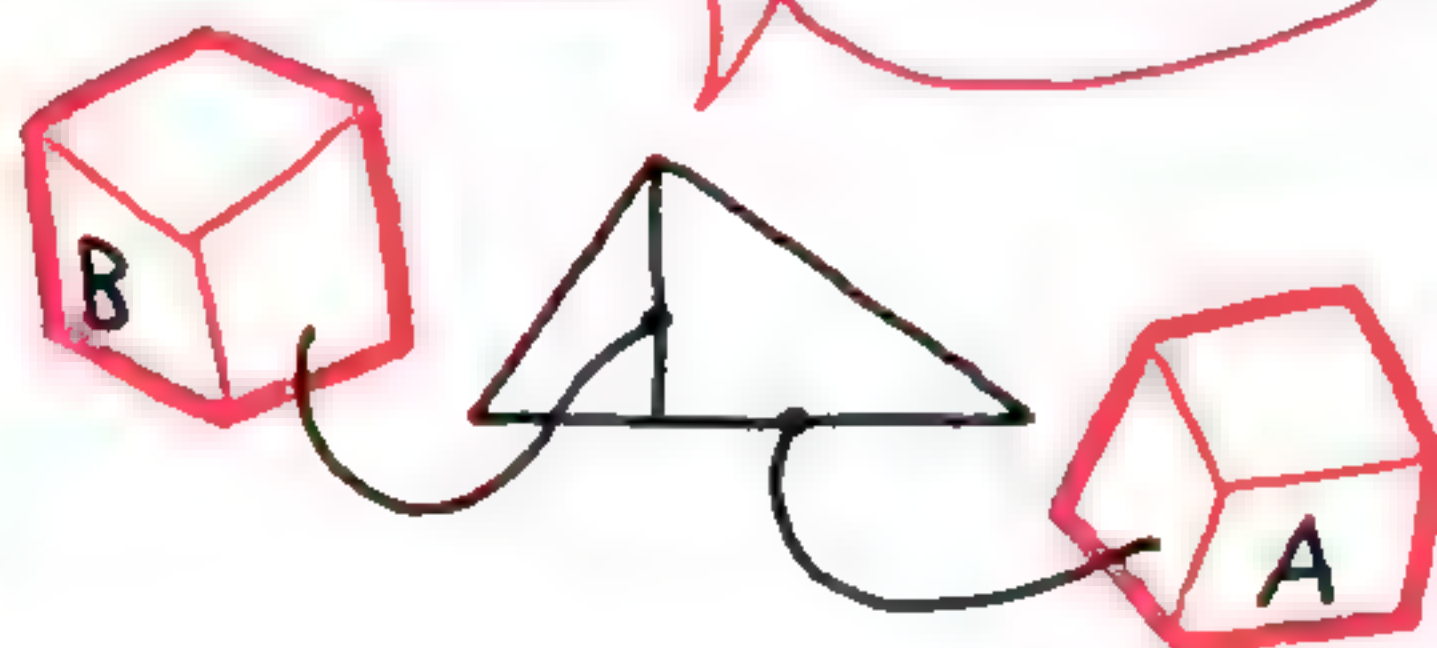
◇……………………………処理内容を分けるための条件を書き込む

では、コンピュータの仕事をフローチャートにしてみよう。

ヒントだよ

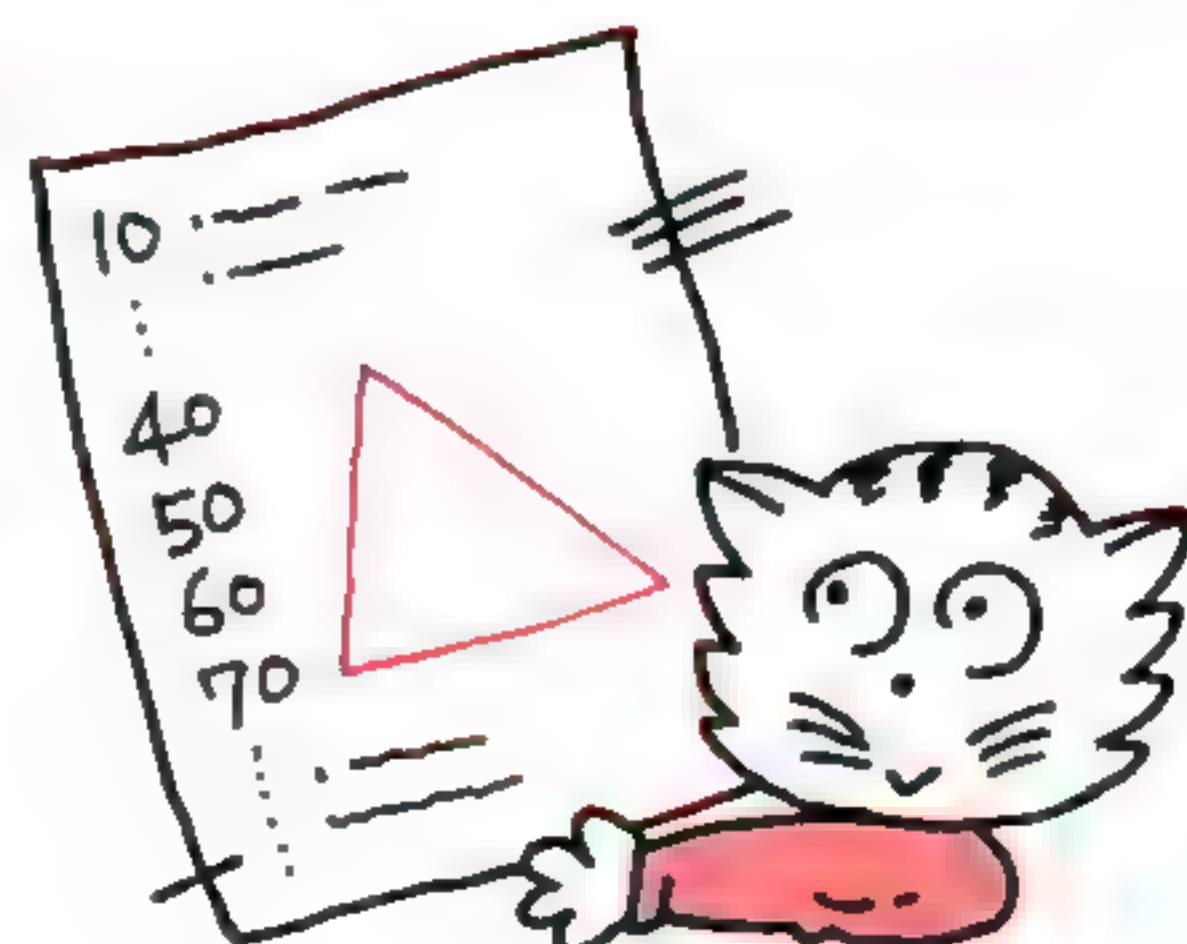
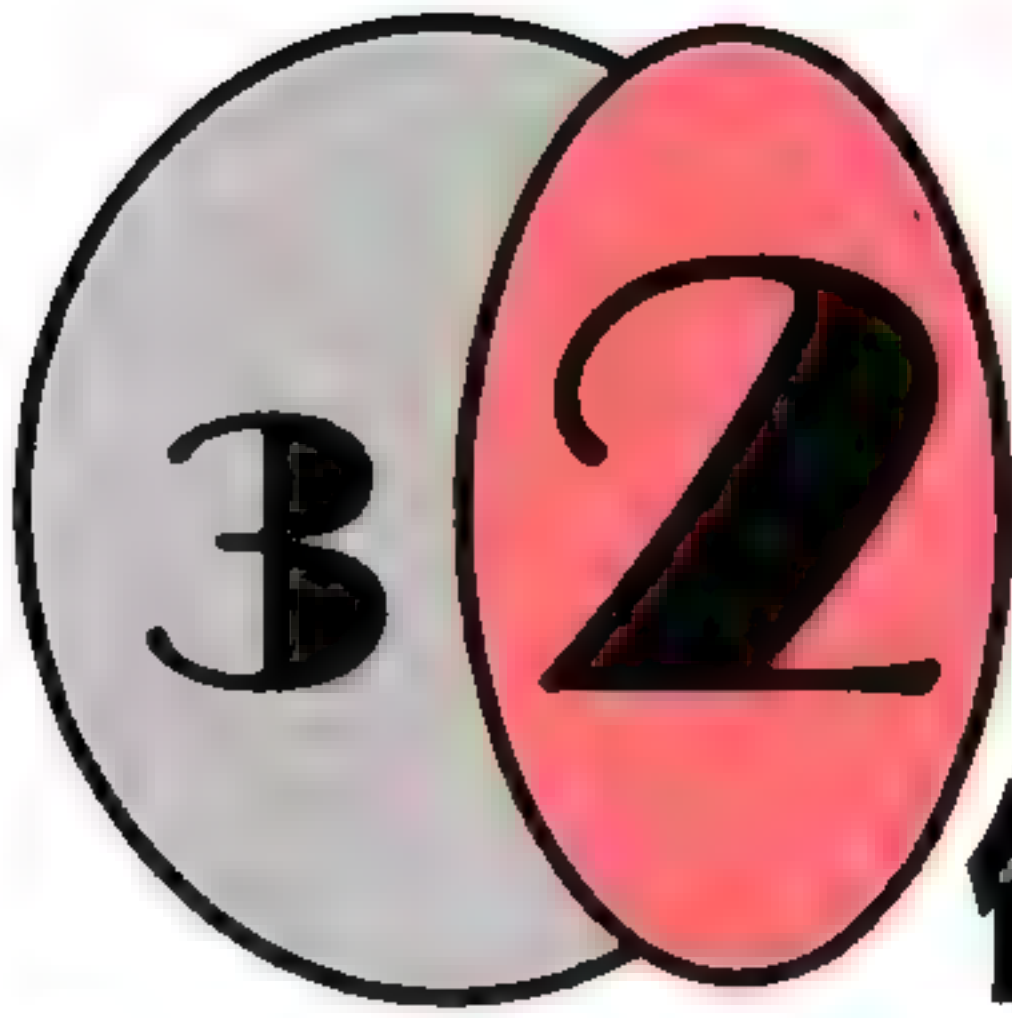
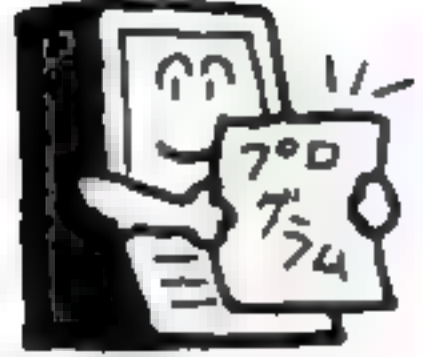
三角形の底辺の長さを変数A、高さを変数B、面積を変数Cにして、プログラミングしてごらん。むずかしいプログラムじゃないから、きみにもできるはずだ

	START
INPUT命令	底辺を入力
INPUT命令	高さを入力
計算式	面積計算
PRINT命令	結果を表示
	END

スタートマークとエンドマークは忘れずに

フローチャートにBASIC命令をあてはめてみよう。

```
10 CLS
20 LOCATE 5,3
30 PRINT"ｻﾝｶｸｹｲ ﾉ ﾒﾝｾｷ"
40 INPUT"ﾃｲﾍﾝ=";A ←――― 底辺の長さを入力
50 INPUT"ﾀｶｻ =";B ←――― 高さを入力する
60 C=A*B/2 ←――― 三角形の面積を計算する
70 PRINT"ﾒﾝｾｷ ﾊ ";C ←――― 結果を表示する
```

三角形の面積を計算するプログラムは、40行から70行までだということは、わかるね。

LOCATE命令とPRINT命令で、何のプログラムかわかるように、タイトルをつけておいたよ。これで完ぺきだね。

仕事の流れが違うとき

IF～THENのフローチャート

きみがキーボードから入力した数が、ぐう数かき数かをコンピュータが区別するプログラムを作ってみよう。

コンピュータの仕事を大きく分けてみると、

1．きみがキーボードから数を入力する
2．ぐう数か、き数かを区別する
3．ぐう数のときには「グウスウ」、き数のときには「キスウ」と画面表示する

これをフローチャートに書いてみるわけだが、その前に、コンピュータがどうやってぐう数とき数を区別するのかを説明しておこう。

ぐう数ってどんな数？

2、4、6、8、10、12………これらはみんなぐう数だね。3、5、7、9、11……はき数だ。

では、ぐう数とはどんな数をいうのかな？　考えてみよう。

ぐう数とは、2で割り切れる数だといえないかな？　逆に、き数は2では割り切れず、余りが出る数だね。

グウスウ

キスウ

コンピュータがぐう数かき数かを区別するのに使う記号がMODだ。MODはわり算をしたときの余りの数を出すときの記号なんだ。だから、

A　MOD　2＝0

という式だったら、Aの数を2で割ったときの余りは0ということだ。余りが0なら、Aはぐう数だね。余りが1なら、Aはき数だ。

これがわかれば、プログラムは作れるよ。さあ、フローチャートを書いてみよう。

Aに6を入れてみる

6　MOD　2

6÷2は、わり切れるから余りの数は

6　MOD　2＝0

だからぐう数だね

●ぐう数か、き数かを区別するフローチャート

START

キーボードから数を入力 ---------- INPUT命令だ

入力した数はぐう数か？

YES → “グウスウ”と画面に表示する

NO → “キスウ”と画面に表示する

END

変数Aに入力した数を入れて、それをコンピュータが区別するんだ

1 2 3 4 5‥

ここからIF～THEN命令を使うぞ

IF～THEN

2

プログラム

表示はPRINT命令だね

グウスウ

```
10 INPUT A
20 IF A MOD 2=0 THEN PRINT"グウスウ"
:ELSE PRINT"キスウ"
30 END
```
①

```
10 INPUT A
20 IF A MOD 2=0 THEN 50
30 PRINT"キスウ"
40 GOTO 60
50 PRINT"グウスウ"
60 END
```
②

プログラムは2通り作れるよ。20行がポイントだ。

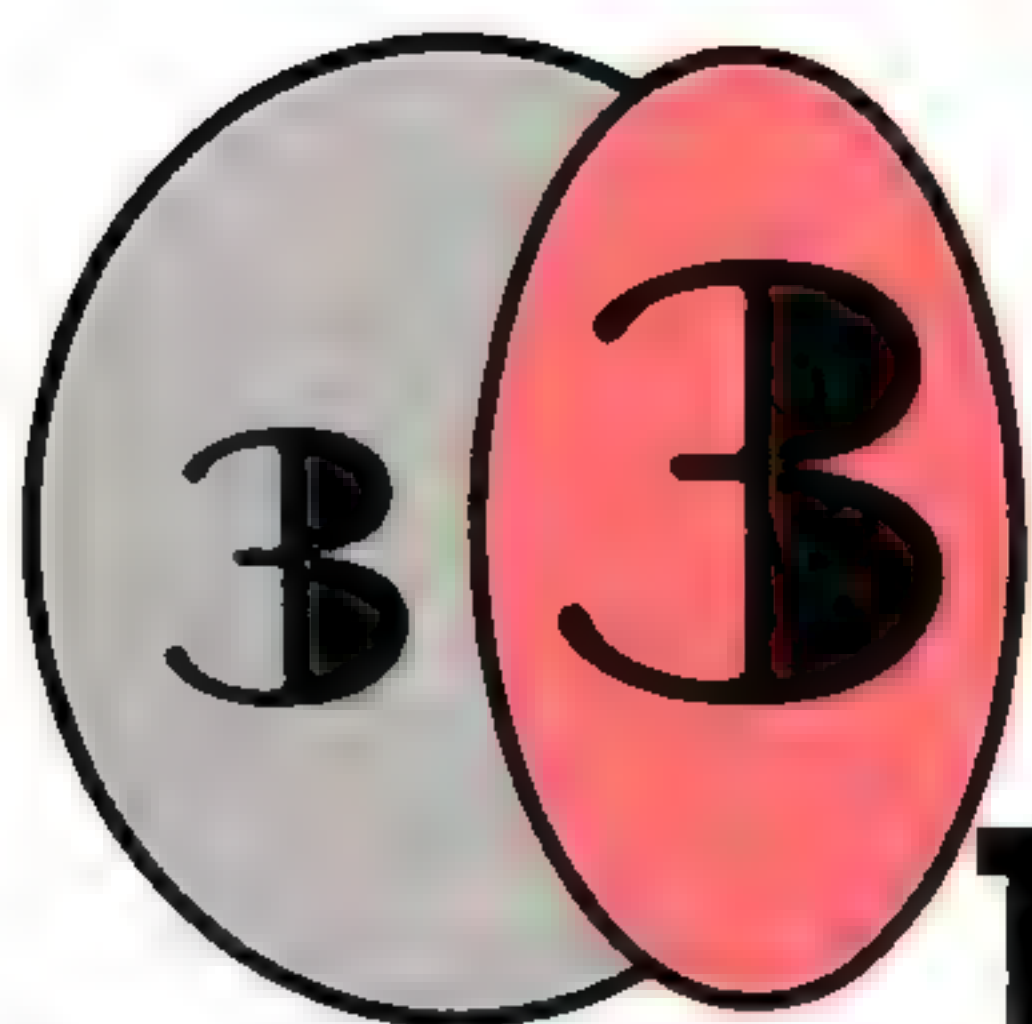
①もしAに代入した数がわり切れる数なら〝グウスウ〟と表示しなさい。

②もしAに代入した数がわり切れる数なら50行へ行きなさい。NOなら30行だ。

IF～THEN～ELSE命令は◇（ひし形）で囲む

IF～THEN～ELSE命令は、いろいろな条件によってコンピュータが処理する仕事の内容が違うときに使う命令だ。

どの条件のときにどんな仕事を実行するか、条件が合わないときはどういう仕事をするかを、それぞれ分けて書いてやらなければならない。

ひし形の囲みの中には？マークを書いておくこと。また、条件が合ったときは（YES）、合わなかったときは（NO）と書いて、それぞれの仕事の行き先を矢印で結ぶ。

1から100までのたし算

FOR～NEXTかIF～THENで

1から100までの数をたし算するといくつになるかという問題だ。1＋2＋3＋4＋5…………＋98＋99＋100＝？　きみが計算すると何分かかるかな？　コンピュータならアッというまだぞ。ただし、計算するためのプログラムを作らないとね。

1＋2＝3
3＋3＝6
6＋4＝10
フゥ～。
たいへんだなぁ

このたし算の考え方を説明しておこう。

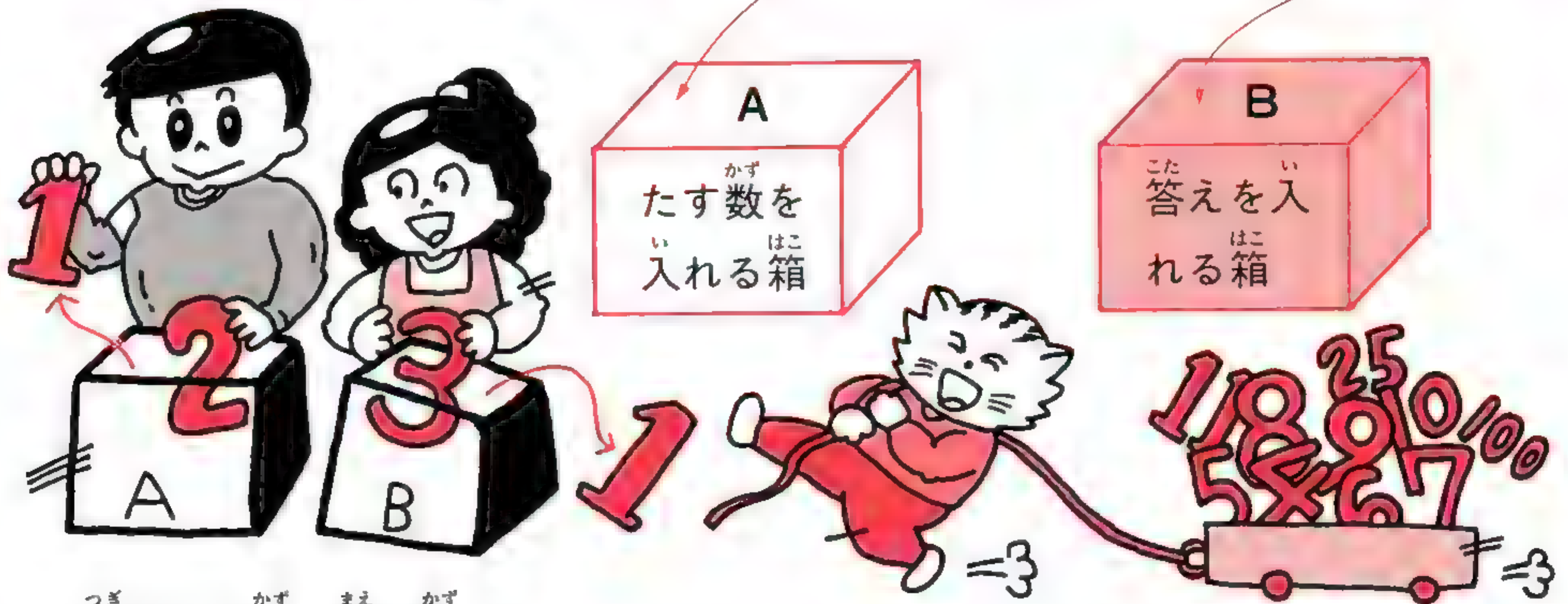
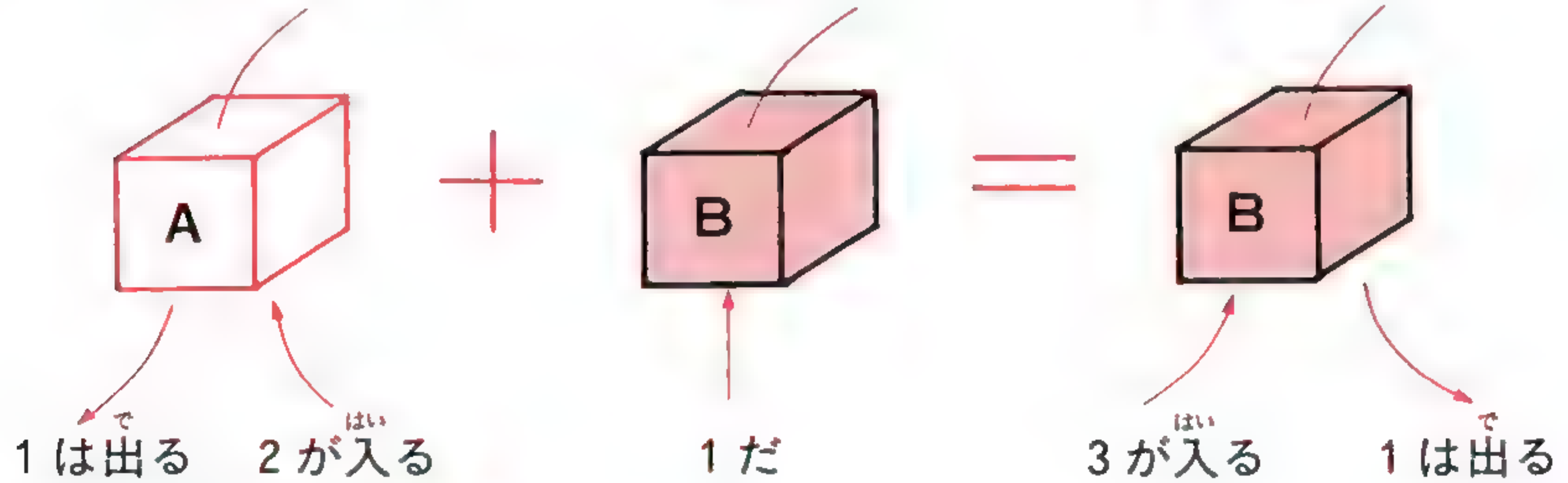
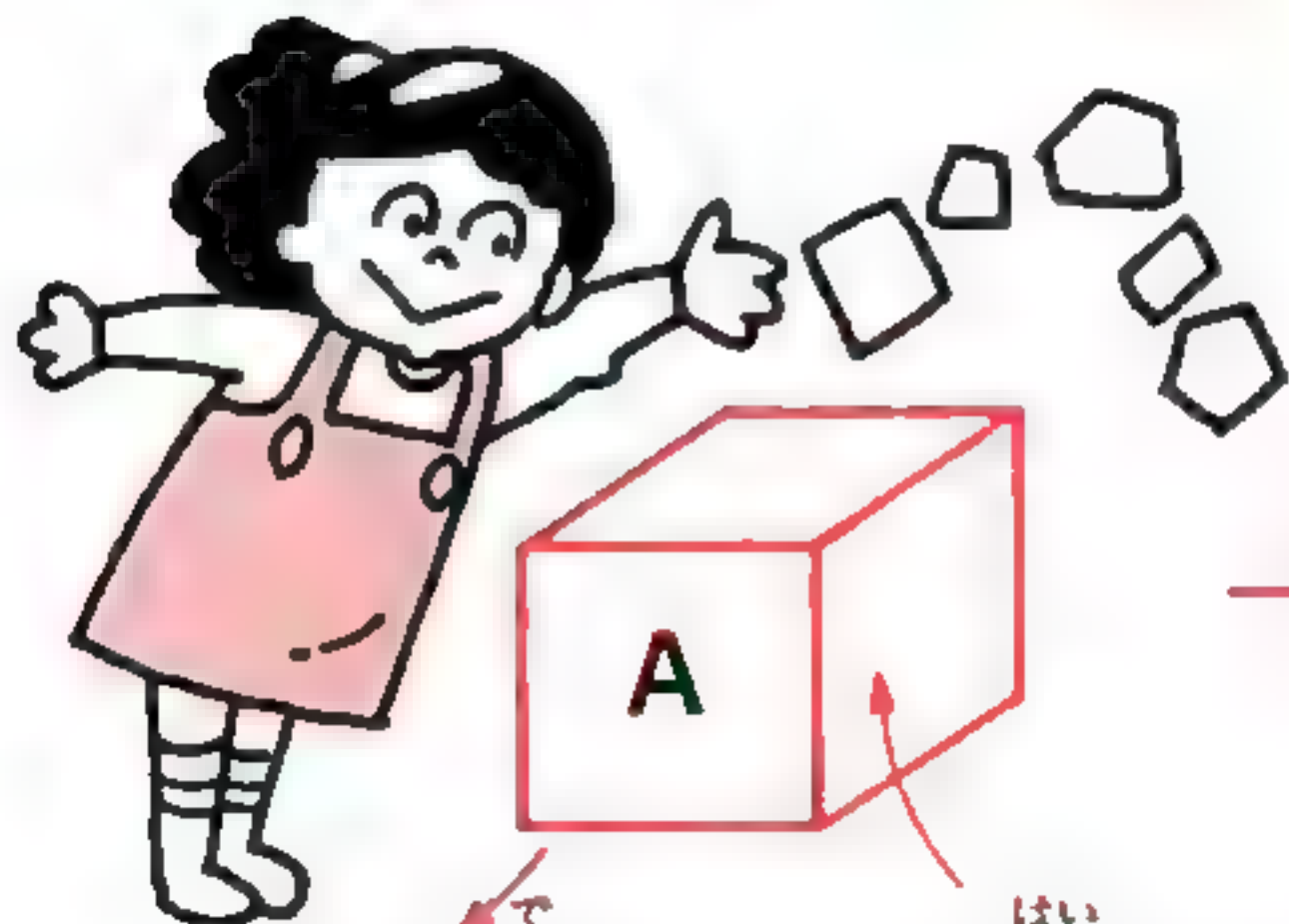
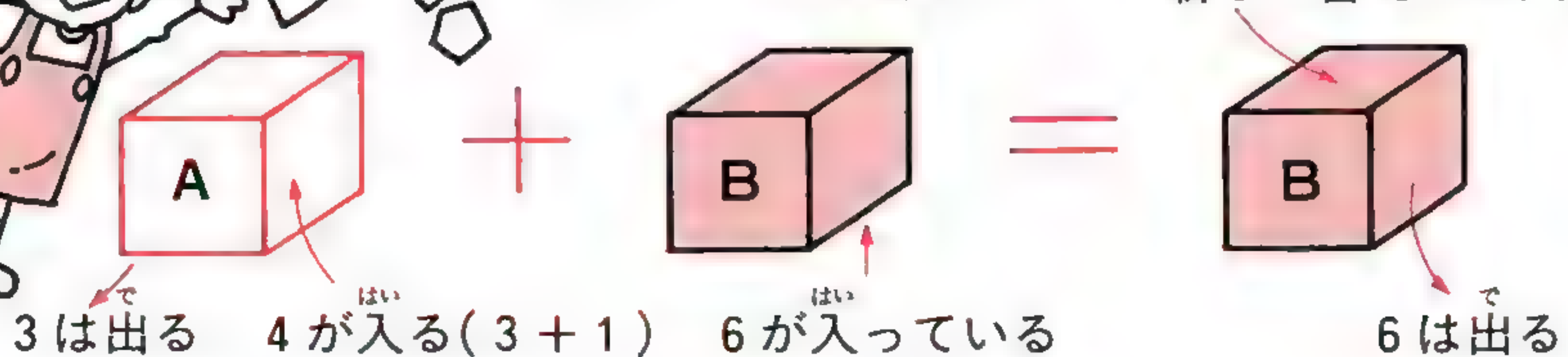
2は1より1多い数。3は2より1多い数。4は3より1多い数だね。つまり、前の数よりも1多い数を順番にたしていけばいいわけだ。

そこでまず、A、Bという箱を2つ用意しよう。Aの箱には次にたす数を入れる。Bの箱はたし算の結果を入れる箱としよう。

次のページに進む前に、自分で紙に箱を書いて考えてごらん

この箱は変数だから、新しい数を入れると前の数は出てしまう性格があったね。これを利用しようというわけだ。どんどん入ったり出たりして、Bに100までたした数が入ったら計算はおわり。

このプログラムは、IF～THEN命令を使っても作れるし、FOR～NEXT命令でも作れるから、2通りの作り方を説明しておこう。

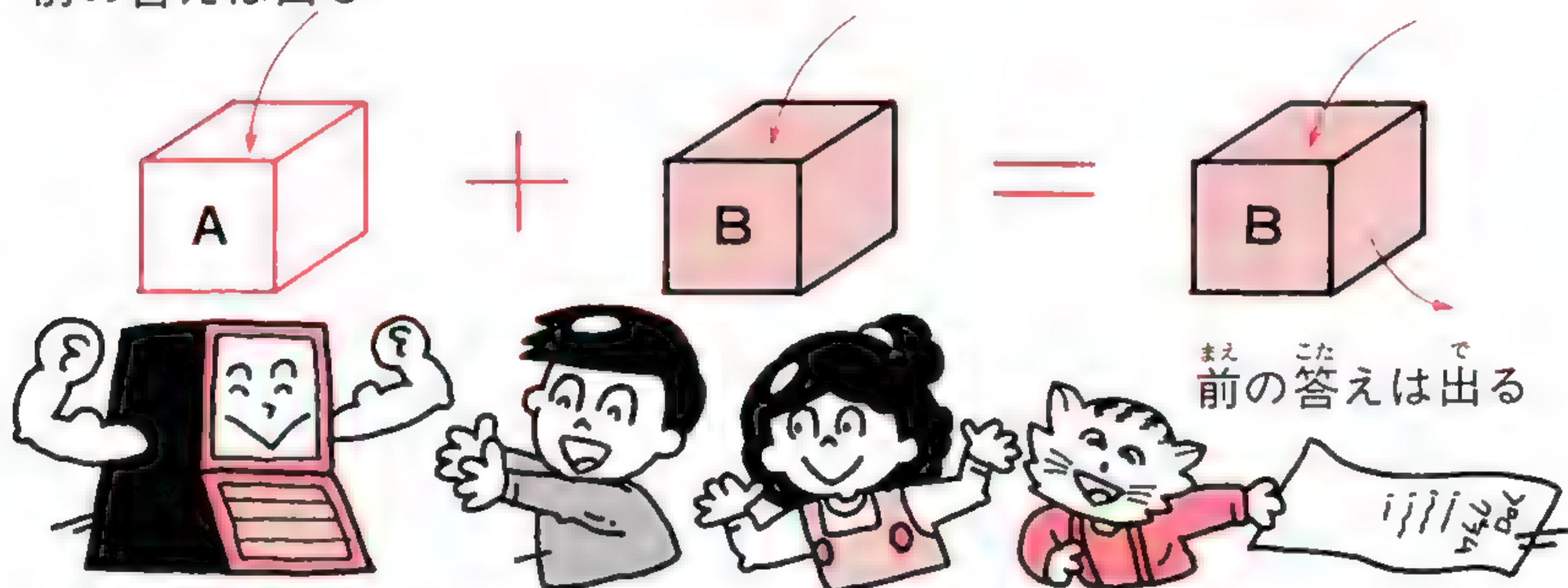
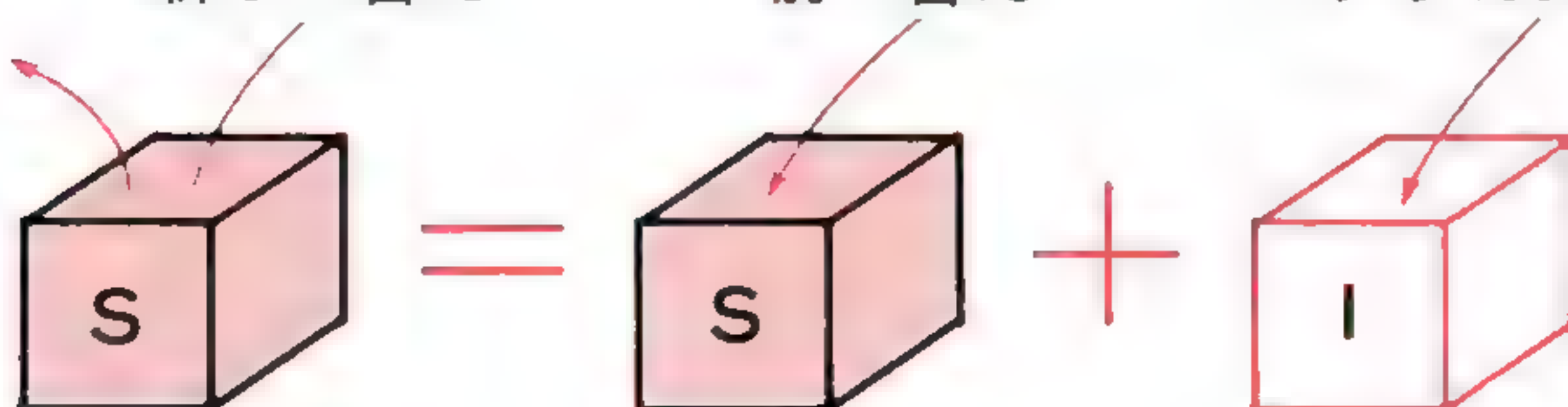
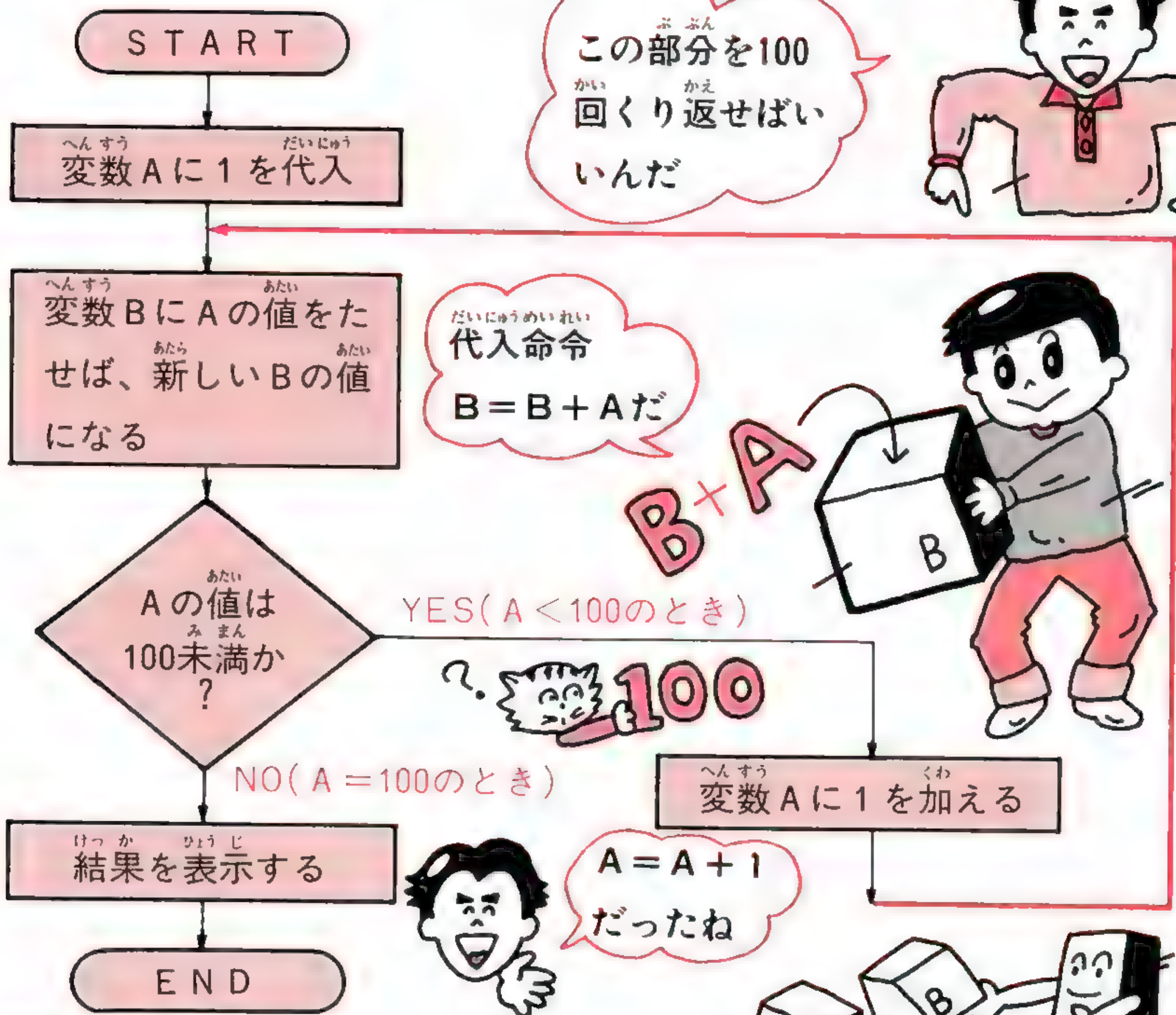
●変数を使って１から100までのたし算をする

＝（イコール）は代入する記号
だったことを思い出してね

A＋1が100になった
99＋1ということだね

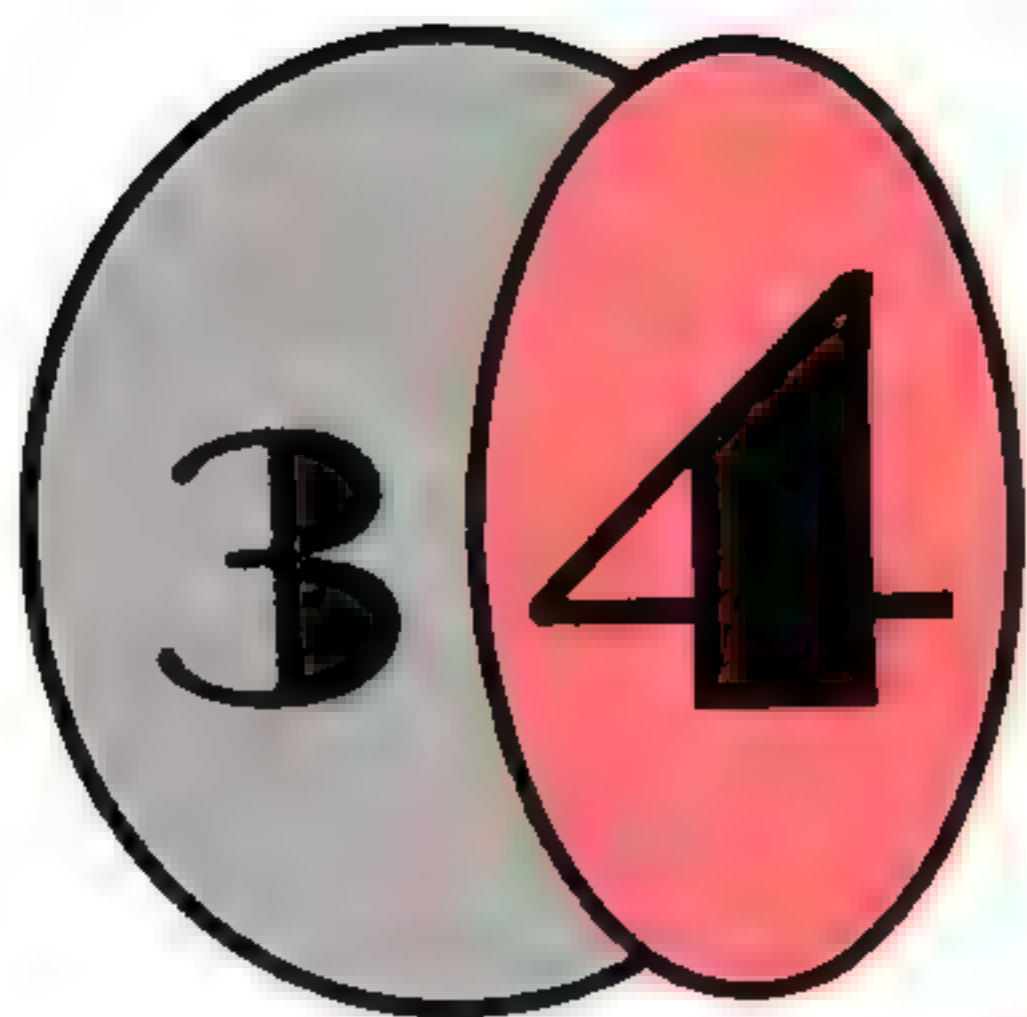
1から99までの数
をたした答え

1から100までたし
た答えが入った

前の答えは出る

前の答えは出る

FOR～NEXTなら1から100までをどんどん入れる

I＝1　TO　100とすれば
1から100までどんどん入
っていく。い
ちいち1をた
さなくていい。

さあ、フローチャートだ！
START
変数Aに1を代入
変数BにAの値をたせば、新しいBの値になる
Aの値は100未満か？
YES（A＜100のとき）
NO（A＝100のとき）
変数Aに1を加える
結果を表示する
END
この部分を100回くり返せばいいんだ
代入命令 B＝B＋Aだ
B+A
B
100
A＝A＋1 だったね
A
B
IF～THEN命令を使ったプログラム
10 A=1
20 B=B+A
30 IF A<100 THEN A=A+1:GOTO 20
40 PRINT B
50 END
①
A＜100の意味は、Aが100より小さい数ということさ
FOR～NEXTを使ったプログラム
10 FOR I=1 TO 100
20 S=S+I
30 NEXT I
40 PRINT S
50 END
②
A＜100
FOR～NEXT
FOR～NEXTだと、ずいぶんかんたんになるなあ

サイコロころころ

パソコンに数を作らせる

RND関数とINT関数●2つそろわなければ整数は作れない

キーボードから数を入力するときは、1でも3でも10でも、好きな数を入力することができるね。

では、コンピュータに数を出させるにはどうしたらいいのだろう。

コンピュータに数を作らせるのが、RND関数だ。関数などという名前はおぼえなくてもいいけれど、次の基本だけはおぼえておこう。

RND（1）

これは、コンピュータにでたらめな数を発生させる命令だ。

次のプログラムを実行してごらん。

```
10  CLS
20  PRINT  RND（1）
30  GOTO  20
```

画面には、どんどん数字がならんでいくぞ。この数はまったくでたらめに作られているんだ。

RND関数というのはでたらめの数を作る働きをするんだよ。

関数とは、こちらからデータを与えるとそれに対して計算や操作を行なって、その結果を返してくれるものだ

関数には便利なものがたくさんある。少しずつ勉強していくといい

●0以上1未満の数を発生させてから

PRINT RND（1）

RND（1）は、0より大きくて1より小さい数をでたらめに発生させるんだ。PRINT命令をつけて、画面に表示させてみよう。

このでたらめに発生する数を、コンピュータでは「乱数」というんだ

0より大きくて1より小さい数というのはどんな数だ？　そう、小数点付きの0.12345…………という数だね。

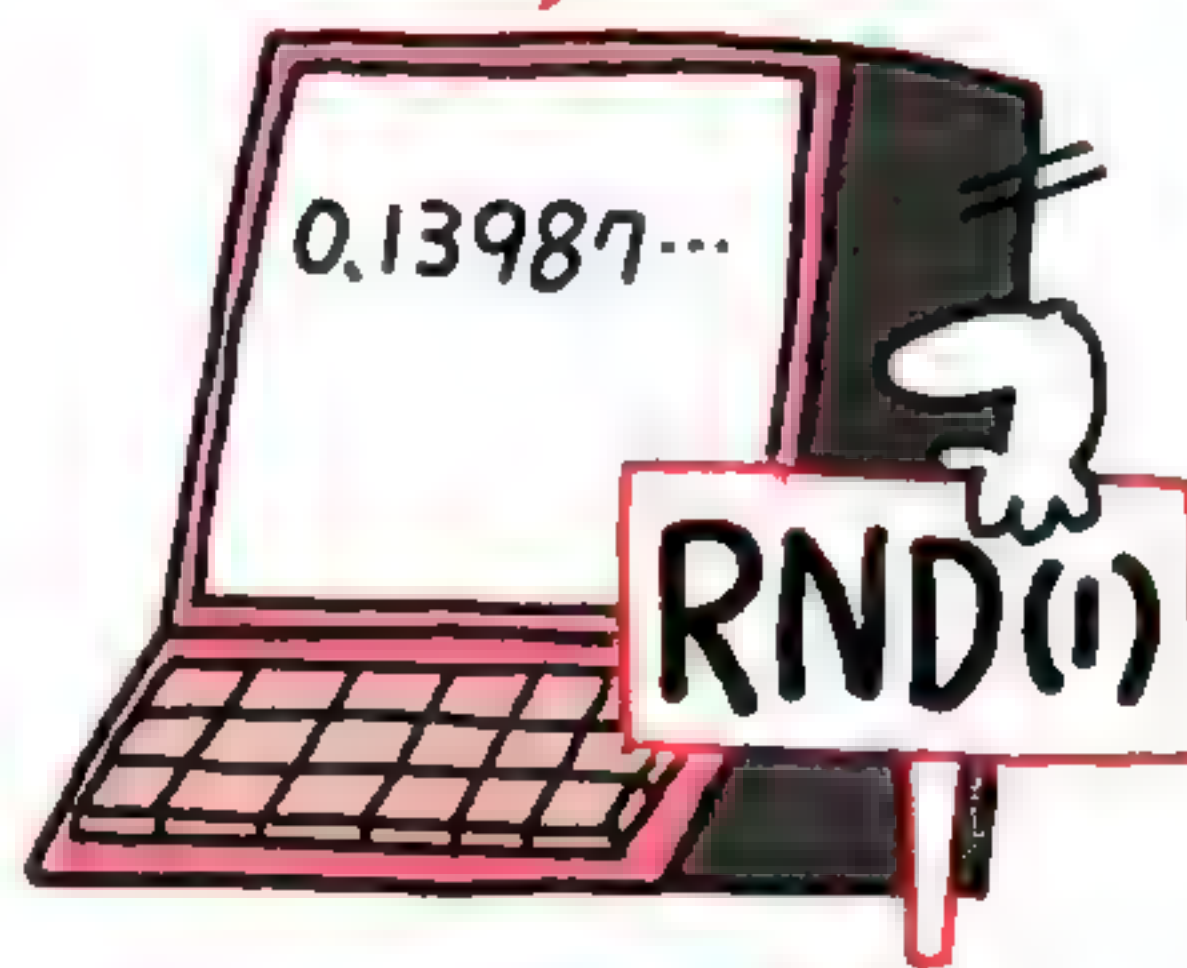
0より大きいのだから 0.0000001……でもいい。また、1より小さくて1に近い数は、0.99999……だ。RND（1）は、この0.△×○△××……という数をでたらめに発生させているが、これより大きな数を発生させるには、どうしたらいいか？

●大きくしたい数だけかける

1よりも大きな数を作りたければ、その数だけかければいいよ。

たとえば、0より大きくて10より小さい数なら、

RND（1）*10

これで、0より大きくて10より小さい数がどんどんできてくる。PRINTで表示してごらん。

小数点以下はとってしまう

小数点付きの数を整数になおすINT文

RNDで数字のもとはできた。でもまだ小数点以下の数がゴチャゴチャとついたままだ。これをとってしまえば、整数ができるね。

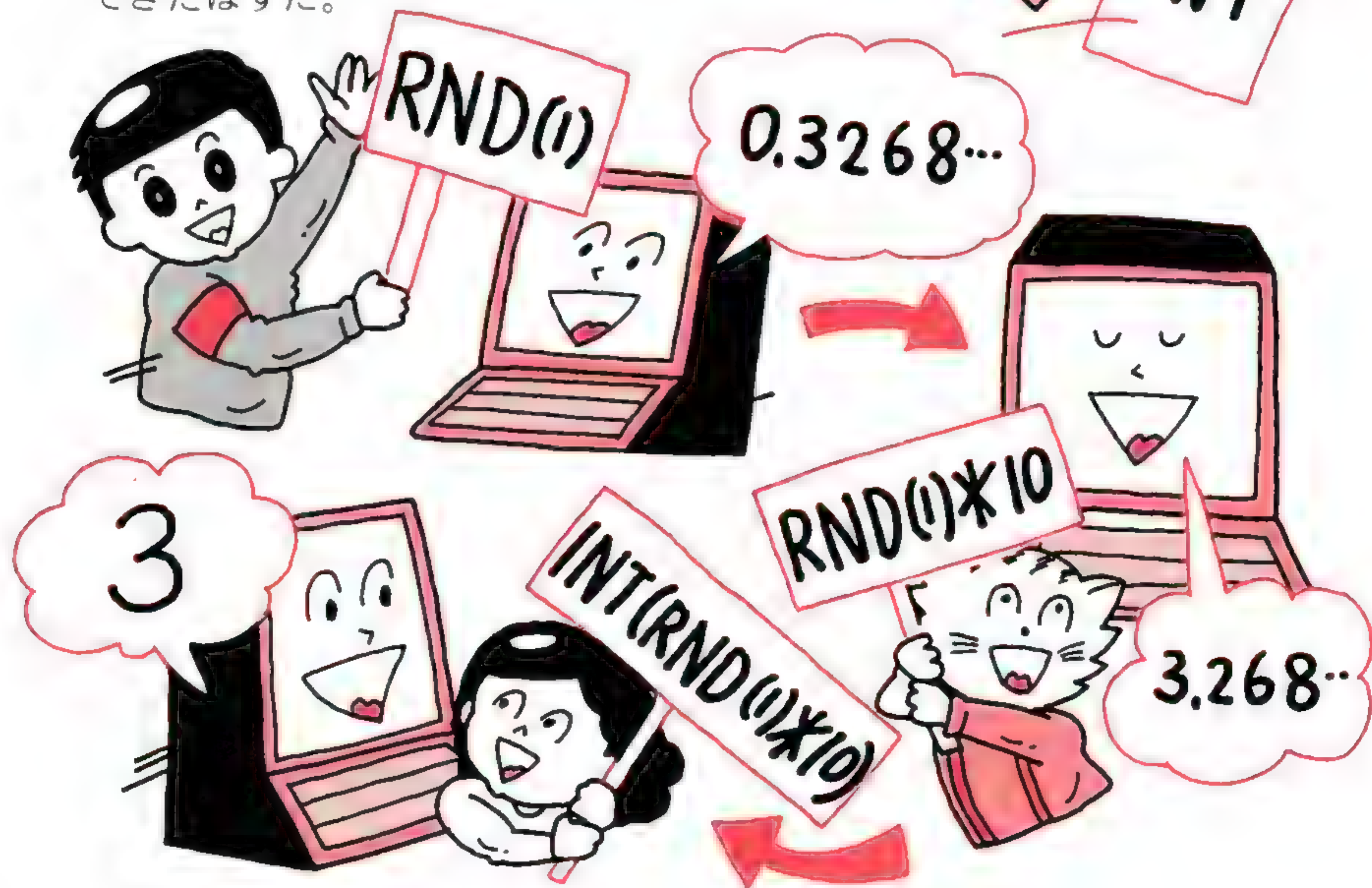
小数点以下をとってしまう命令が、INT文だ。書き方は、こうなる。

```
INT (RND (1) *10)
```

画面に表示させるには、

```
PRINT  INT (RND (1) *10)
```

RETURNキーを押して実行してごらん。

さあ、これで、0より大きくて10より小さい整数、つまり0から9までの数がどんどん画面に出てきたはずだ。

整数になったけど、この数は乱数だから順番には出てこないよ

1から6までの数を作る

きみもよく知っているサイコロの目を出してみよう。1から6までの数だ。

まず、0より大きくて6より小さい数を作る。

RND（1）＊6

これで0より大きくて6より小さい数ができる。これでは0から5までの数しかできない。1より大きく7より小さい数を作らなければならないから、これに1をたしてやる。

RND（1）＊6＋1

まだ小数点以下の数がついたままだから、INT命令でとると、

INT（RND（1）＊6＋1）

この命令を与えてやると、コンピュータの中で1から6までの乱数が作られる。

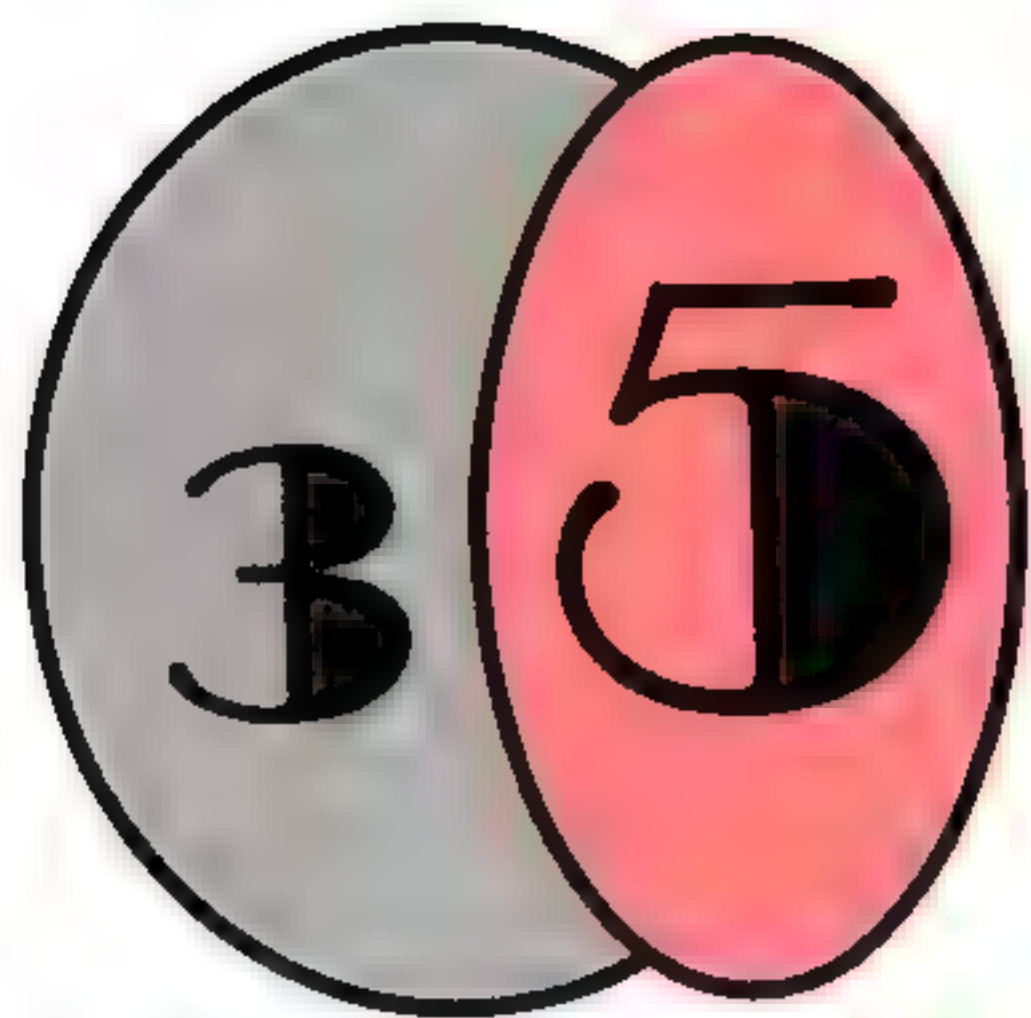
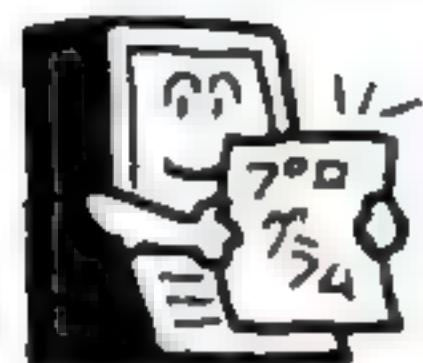
RNDとINTは、数に関係するいろいろなところで使われているよ。何回ゲームができるとか、キャノン砲を何発出すようにするとか、ネ

● 1から6までの数を作るプログラム

①
```
10 CLS
20 A=INT(RND(1)*6+1)
30 PRINT A;
40 GOTO 20
```

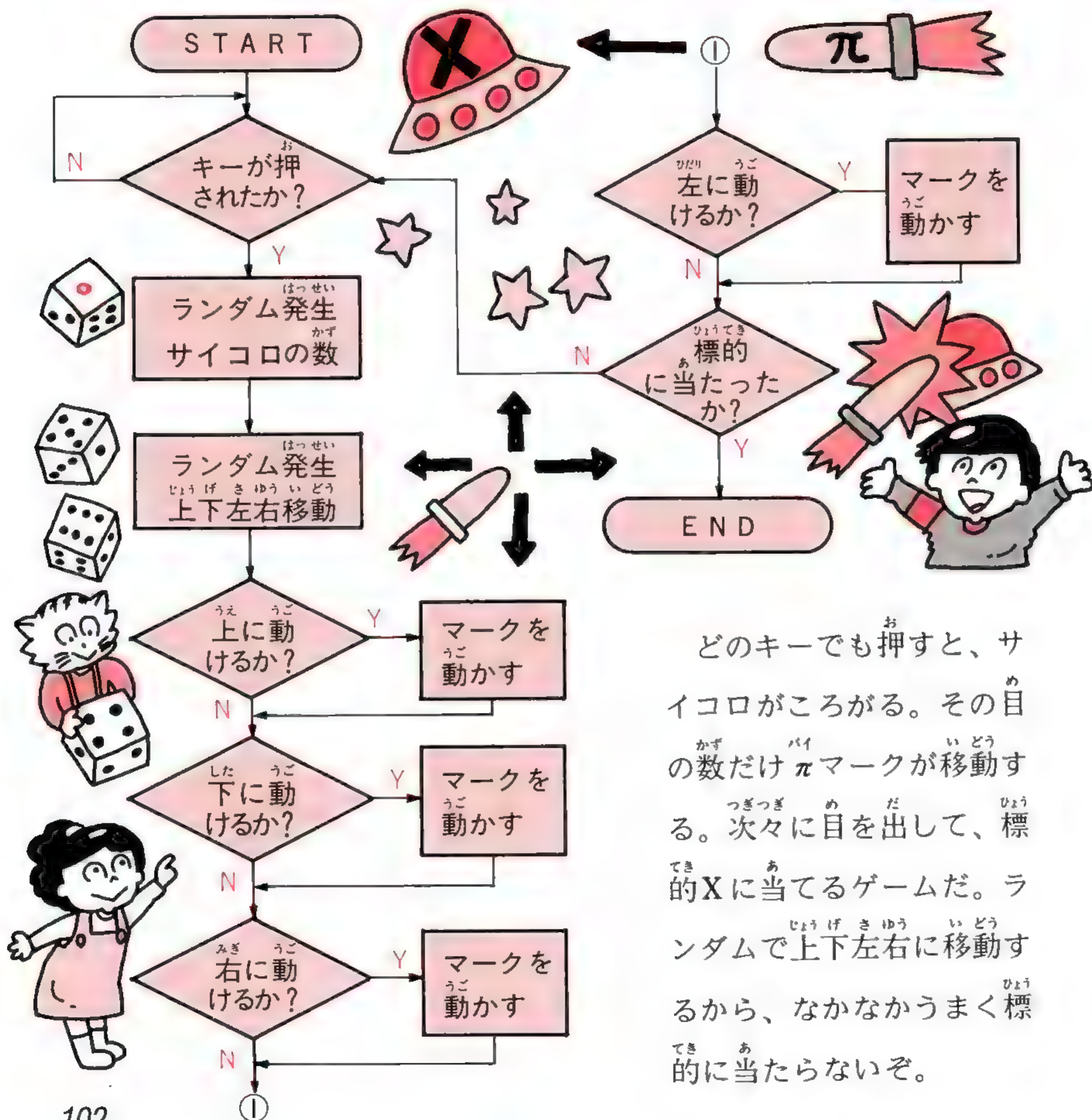
● 1から6までの数をあてるゲーム

キーボードから、きみの好きな数を入力してみよう。コンピュータの持っている数と、きみの入力した数があえば、"アタリ"と表示されるよ

②
```
10 A=INT(RND(1)*6+1)
20 INPUT "1ｶﾗ 6ﾏﾃﾞﾉ ｷｰｦ ｵｼﾃｸﾀﾞｻｲ";B
30 IF A=B THEN PRINT"ｱﾀﾘ!!"
   ELSE PRINT"ﾊｽﾞﾚ!!"
40 GOTO 10
```

サイコロ応用ゲーム

πマークを移動して標的に当てる

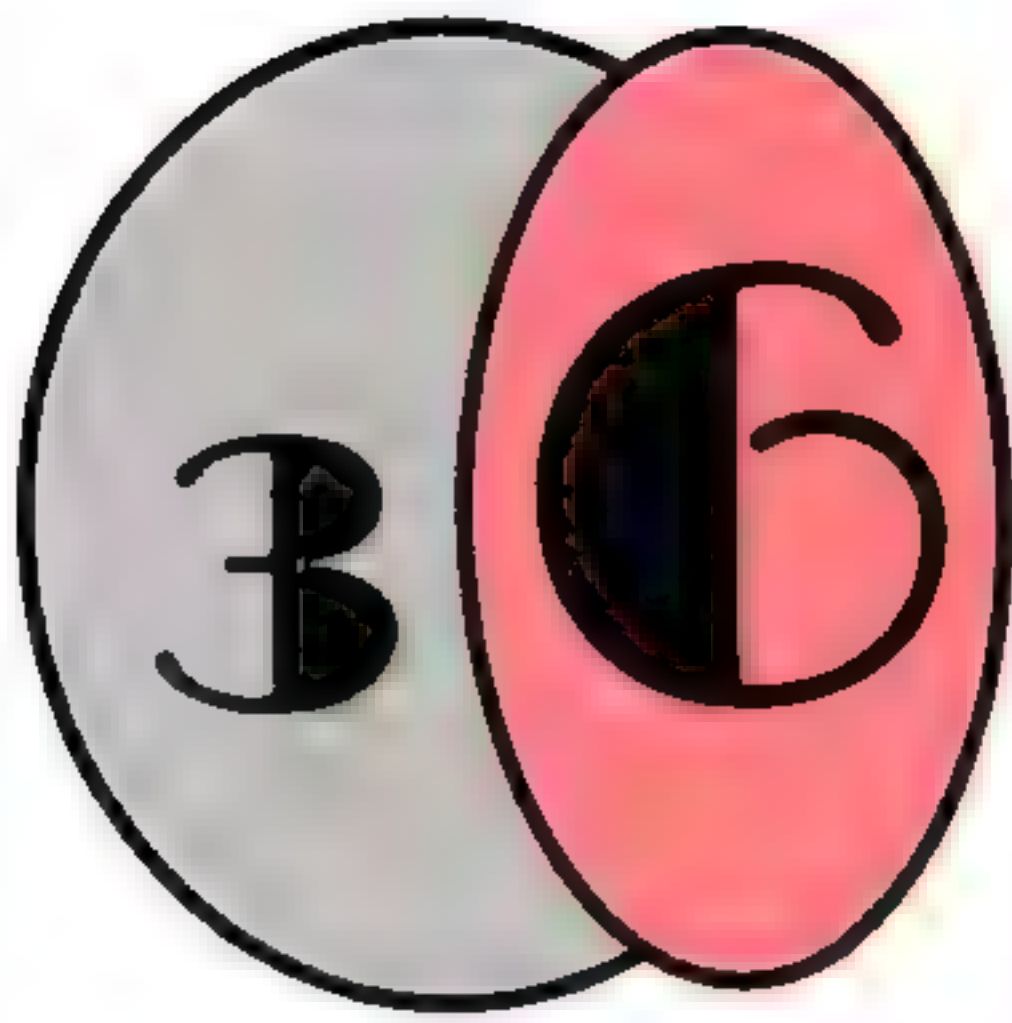
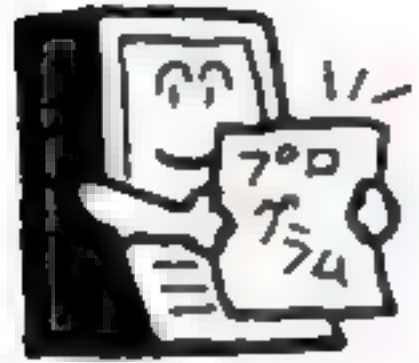
まず、フローチャートからだ。

START
→ キーが押されたか？ （N → キーが押されたか？へ戻る）
Y → ランダム発生 サイコロの数
→ ランダム発生 上下左右移動
→ 上に動けるか？ Y → マークを動かす
N → 下に動けるか？ Y → マークを動かす
N → 右に動けるか？ Y → マークを動かす
N → ①

① → 左に動けるか？ Y → マークを動かす
N → 標的に当たったか？ （N → キーが押されたか？へ）
Y → END

どのキーでも押すと、サイコロがころがる。その目の数だけπマークが移動する。次々に目を出して、標的Xに当てるゲームだ。ランダムで上下左右に移動するから、なかなかうまく標的に当たらないぞ。

■ゲームプログラム　サイコロコロコロ

```
10 CLS
20 SCREEN 0,0
30 WIDTH 40:KEY OFF
40 FOR Y=1 TO 3
50 LOCATE 0,Y
60 FOR X=1 TO 40
70 PRINT"♥";
80 NEXT X,Y
90 PRINT"┌────────────────────";
100 PRINT"────────────────────┐"
110 FOR N=4 TO 17
120 PRINT"|                                      |"
130 NEXT
140 PRINT"└────────────────────";
150 PRINT"────────────────────┘"
160 X1=6:Y1=6
170 LOCATE 20,11:PRINT"x"
180 LOCATE X1,Y1:PRINT"π"
190 S=0:LOCATE 12,2
200 PRINT"SAIKORO KOROKORO"
210 K$=INKEY$:IF K$=""THEN 210
220 A=INT(RND(-TIME)*6)+1
230 BEEP:S=S+1
240  IF K$=" " THEN PRINT"GOOD BY!":END
250 X=30:Y=19
260 ON A GOSUB 480,540,600,660,720,780
270 IF S=15 THEN 290
280 GOTO 220
290 C=INT(RND(-TIME)*6)+1
300 LOCATE X1,Y1:PRINT" "
310 IF (C=2)OR(C=3) THEN X1=X1+A
320 IF C=5 THEN X1=X1-A
330 IF C=6 THEN Y1=Y1-A
340 IF (C=1)OR(C=4) THEN Y1=Y1+A
350 IF X1<=2 THEN X1=2
360 IF X1>=37 THEN X1=37
370 IF Y1<=5 THEN Y1=5
380 IF Y1>=18 THEN Y1=18
390 LOCATE X1,Y1:PRINT"π"
400 IF (X1=20)AND(Y1=11) THEN 420
410 S=0:GOTO 210
```

```
420 LOCATE 19,10:PRINT"END"
430 FOR M=1 TO 20
440 BEEP:LOCATE 19,10:PRINT"   "
450 FOR T=1 TO 100:NEXT
460 LOCATE 19,10:PRINT"END"
470 NEXT:END
480 LOCATE X,Y:  PRINT"┌───┐"
490 LOCATE X,Y+1:PRINT"│   │"
500 LOCATE X,Y+2:PRINT"│ ● │"
510 LOCATE X,Y+3:PRINT"│   │"
520 LOCATE X,Y+4:PRINT"└───┘";
530 RETURN
540 LOCATE X,Y:  PRINT"┌───┐"
550 LOCATE X,Y+1:PRINT"│o  │"
560 LOCATE X,Y+2:PRINT"│   │"
570 LOCATE X,Y+3:PRINT"│  o│"
580 LOCATE X,Y+4:PRINT"└───┘";
590 RETURN
600 LOCATE X,Y:  PRINT"┌───┐"
610 LOCATE X,Y+1:PRINT"│o  │"
620 LOCATE X,Y+2:PRINT"│ o │"
630 LOCATE X,Y+3:PRINT"│  o│"
640 LOCATE X,Y+4:PRINT"└───┘";
650 RETURN
660 LOCATE X,Y:  PRINT"┌───┐"
670 LOCATE X,Y+1:PRINT"│o o│"
680 LOCATE X,Y+2:PRINT"│   │"
690 LOCATE X,Y+3:PRINT"│o o│"
700 LOCATE X,Y+4:PRINT"└───┘";
710 RETURN
720 LOCATE X,Y:  PRINT"┌───┐"
730 LOCATE X,Y+1:PRINT"│o o│"
740 LOCATE X,Y+2:PRINT"│ o │"
750 LOCATE X,Y+3:PRINT"│o o│"
760 LOCATE X,Y+4:PRINT"└───┘";
770 RETURN
780 LOCATE X,Y:  PRINT"┌───┐"
790 LOCATE X,Y+1:PRINT"│ooo│"
800 LOCATE X,Y+2:PRINT"│   │"
810 LOCATE X,Y+3:PRINT"│ooo│"
820 LOCATE X,Y+4:PRINT"└───┘";
830 RETURN
```

さんすうドリルプログラム

10問中いくつあたったか？

かんたんな計算問題のドリルを作ってみよう。問題はひと桁のかけ算だ。仕事の流れを考えてから、フローチャートを書いてみよう。

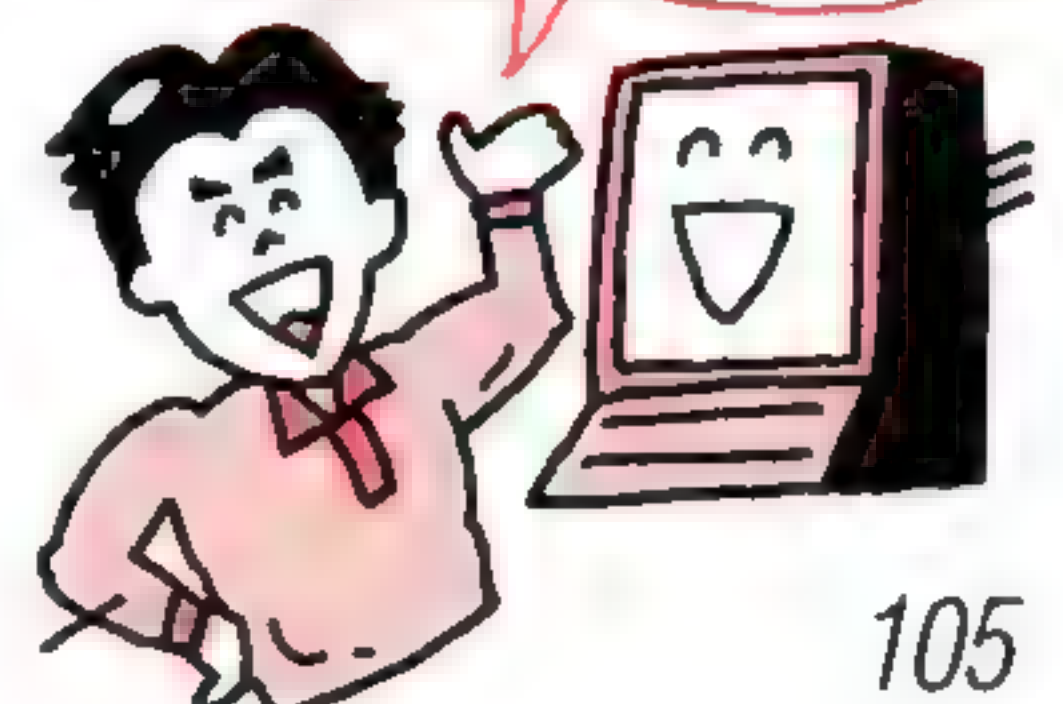
プログラミングの流れ

1．コンピュータが問題を作って、画面に表示する。
2．その問題の答えを、きみがキーボードから入力する。
3．答えがあっているかどうかを、コンピュータが判断する。答えがあっていれば○、まちがっていれば×を表示する。
4．問題が10問になると、終わる。

フローチャートを書くまえに、どんな命令を使ったら、このプログラムが作れるのかを自分で考えてみよう

フローチャートは、これを図に直せばいいだけだ。それほどむずかしいプログラムではないから、先に進む前に、トライしてみてもいいよ。

ポイントとなる命令は、問題を作るために数を発生させるRND、INTと、アタリ、ハズレを判定するIF～THEN命令みたいだね。

乱数を使えば、どんな問題が出るのかきみにもわからないぞ

さんすうドリルのフローチャート

10問の問題を発生させるループ。FOR～NEXTだ。10問出し終わったら、ENDだ。

ひと桁の計算問題だから、1から9までの数だ。作ってみよう。

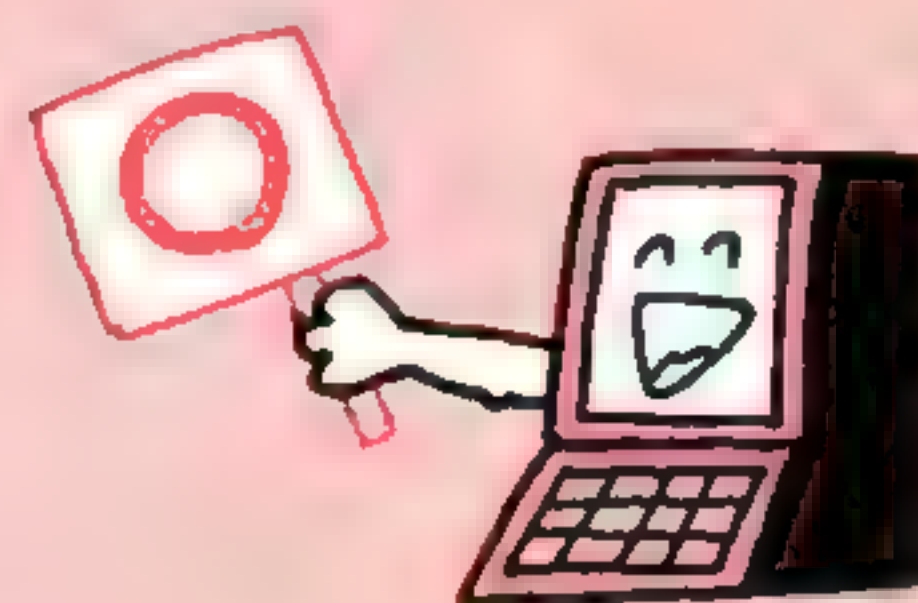
INPUT命令だね。

IF～THENが出てきたぞ。

このフローチャートで、たし算、ひき算、かけ算、わり算の4種類の計算問題を作れる。桁数をふやせば、むずかしい問題も出せるぞ

●さんすうドリルのプログラム

```
10 CLS
20 SCREEN 0,0
30 WIDTH 40
40 KEY OFF
50 LOCATE 15,0:PRINT"ｶｹｻﾞﾝ"
60 FOR N=1 TO 10
70 X=INT(RND(-TIME)*9+1)
80 Y=INT(RND(-TIME)*9+1)
90 LOCATE 8,1+N*2
100 PRINT X;"*";Y;"="
110 LOCATE 17,1+N*2:INPUT A
120 IF X*Y=A THEN LOCATE 23,1+N*2:S=S+1
130 IF X*Y=A THEN PRINT"O":NEXT:GOTO160
140 LOCATE 23,1+N*2
150 PRINT" X ":NEXT
160 LOCATE 27,18
170 PRINT"10ﾓﾝ ﾁｭｳ "
180 LOCATE 26,19
190 PRINTS;"ﾓﾝ ｱﾀﾘﾃﾞｽ"
200 LOCATE 27,21
210 PRINT"ｷｰ ｦ ｵｽ"
220 Z$=INKEY$:IF Z$="" THEN GOTO 220
230 RUN
```

[illegible]を出して、○×を判定するループ

●プログラムの説明

かんたんにプログラムの説明をしておこう。

10行～50行

初期設定といって、画面表示の準備をする部分だ。20行のSCREEN 0，0で、文字表示用の画面を指定して、WIDTH 40で、横に40字表示できるようにした。

40行の「KEY OFF」は、初めて出てくる命令だね。電源を入れたときの画面には、下のほうに

KEY OFFは、ファンクションキーの内容を画面から消す命令だ

ファンクションキーの内容が表示されているね。これはべつに表示されているままでもかまわないけれど、ないほうが画面が見やすくなるというときには、KEY OFF命令を実行すると消えてしまう。

：（コロン）でつなぐマルチステートメントがいくつか出てくるから注意しよう

60行〜150行

この部分のFOR〜NEXTループで、10回問題を出しているのがわかるかな？

```
60 FOR N=1 TO10
70 X=INT(RND(-TIME)*9+1)
80 Y=INT(RND(-TIME)*9+1)
90 LOCATE 8,1+N*2
100 PRINT X;"*";Y;"="
110 LOCATE 17,1+N*2:INPUT A
120 IF X*Y=A THEN LOCATE 23,1+N*2:S=S+1
130 IF X*Y=A THEN PRINT"o":NEXT:GOTO 160
140 LOCATE 23,1+N*2
150 PRINT "x":NEXT
```

かけ算をする２つの数をX、Yの変数として、１から９までの整数をランダムに発生させているのが70行と80行だ。

ここで（－TIME）という見なれないものが出てきたね。１から９までの整数を発生させるには、

ＩＮＴ（ＲＮＤ（１）＊９＋１）

でも、もちろんいい。

RND（１）だと、あれ、何回か乱数を出していると、同じパターンに。RND（－TIME）なら、まったく不規則だ

ただ、この場合は乱数を発生させるたびに同じような値が出てくるんだ。だから同じ問題が何回も出てきてしまうこともある。

RND（－TIME）とすると、カッコ内の数を自動的に変えて、毎回発生させる乱数を変えられるんだ。だから、いろいろな問題が出題される。

120行と130行はIF文で答えがあたったときの処理だ。X＊Yの答えがAと一致しないときには、140行から150行へ行って、まちがったときの処理、つまり×を画面に表示させるよ。

5+1=6 ○
6+5=9 ×

120行のS＝S＋1はあたったときの回数をたしている。○印を表示してから160行に飛ぶよ。変数Sは190行にあるね

160行～190行

10問中いくつあたったかを画面に表示する。

200行～230行

「キー　ヲ　オス」と画面表示が出たら、キーボードから何でも好きなキーを押せば、もう一度、10問の計算問題がスタートする。

文字をつなげて表示させる；（セミコロン）

100行を見てみよう。Xは70行で発生させた乱数。Yは80行で発生させた乱数だ。；は、文字をつなげて表示するときに使う記号だ。

たとえばXが9、Yが5のときには、9＊5と表示される。

110行では、＝のうしろにカーソルを指定して、INPUTの？マークを表示させて、右の画面のような計算問題ができる。

；を，（コンマ）にかえると、次の文字との間に13文字分の空白をあけてしまう。

たし算、ひき算、わり算のドリル

●たし算のプログラムに変える

100行と120行、130行のアスタリスク（＊）を＋（プラス）の記号に変えればＯＫ。

問題をむずかしくしたければ、70行と80行の９をもっと大きな数に変えればいい。

＊（カケル）を、＋（タス）、－（ヒク）、／（ワル）に変える

●ひき算のプログラムに変える

100行、120行、130行を－（マイナス）に変えるのはたし算と同じだ。ただし、このままだと引く数が引かれる数より大きくなって、－（マイナス）の答えが出ることもあるよ。

マイナスの答えを出したくない場合は、プログラムを１行追加する。80行のあとに、

```
85 IF X<Y THEN 70
```

この意味は、ＹがＸより大きいときは、もう一度、数を発生させるために70行にもどるということだ。追加したら RETURN キーを押すよ。

```
  LOCATE 8,1
.00 PRINT X;"*",
110 LOCATE 17,1+
120 IF X*Y=A THE
130 IF X*Y=A THE
'40 LOCATE 23,1
```

ひき算、わり算は、ちょっと工夫が必要

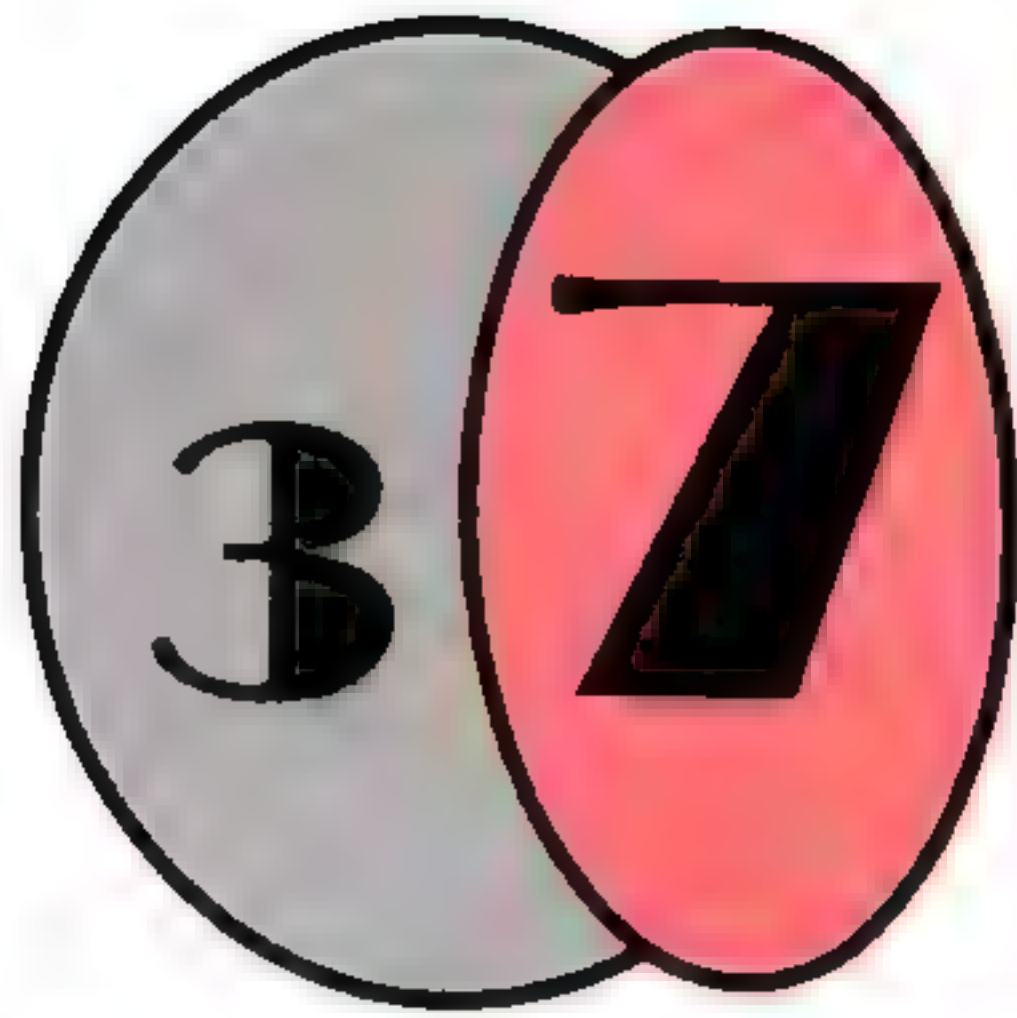
●わり算のプログラムに変える

わり算のプログラムに変えるときには、もう１行プログラムを追加する。86行として、

```
86 IF (X/Y)<>INT(X/Y) THEN 70
```

これは、わり切れる問題を出すためのプログラムで、<>の記号は「等しくない」という意味だ。

追加したら、LIST命令で確認しておこう。

追加したらかならず RETURN キー。LISTして確認することも忘れずに。

プログラムの整理

行を加えたりけずったりしたとき

RENUM命令●プログラムの行番号を整理する

RENUM命令は、行番号をそろえる命令だ。

わり算プログラムに変えたときを考えてみよう。AUTO命令で、10番きざみにきれいに並んでいたプログラムが、途中から変わってしまったね。

もう一度10番きざみの行番号にして整理しよう。

RENUM

とキー入力して、RETURNキーを押す。

画面上のプログラムはなにも変化しないけれど、RENUM表示のすぐ下にＯＫと表示されているはずだ。RENUM命令は、「プログラムの行番号を付け変えろ」という命令だけで、整理されたプログラムを表示させるのは、LIST命令だ。

10番きざみにつけた行番号のあいだには、9個の行番号が使える。追加したあと、RENUMで10番きざみに戻しながら、いくらでもプログラムを追加できるぞ

DELETE命令●プログラムを行番号ごととってしまう

DELETE命令は、プログラムを行番号ごととってしまう命令だ。2～3行消すだけだったら、消したい行番号をキー入力して、RETURNキーを押せば消えてしまう。DELETE命令は、たくさんのプログラムをまとめて消すときに便利なんだ。

一度消すともとにもどらないから慎重に

使い方もいろいろある。[D][E][L][E][T][E]とキー入力したら、[RETURN]キーを忘れずに。終わったら、同じくRENUMで整理して、LIST命令で画面に出して確認しよう。

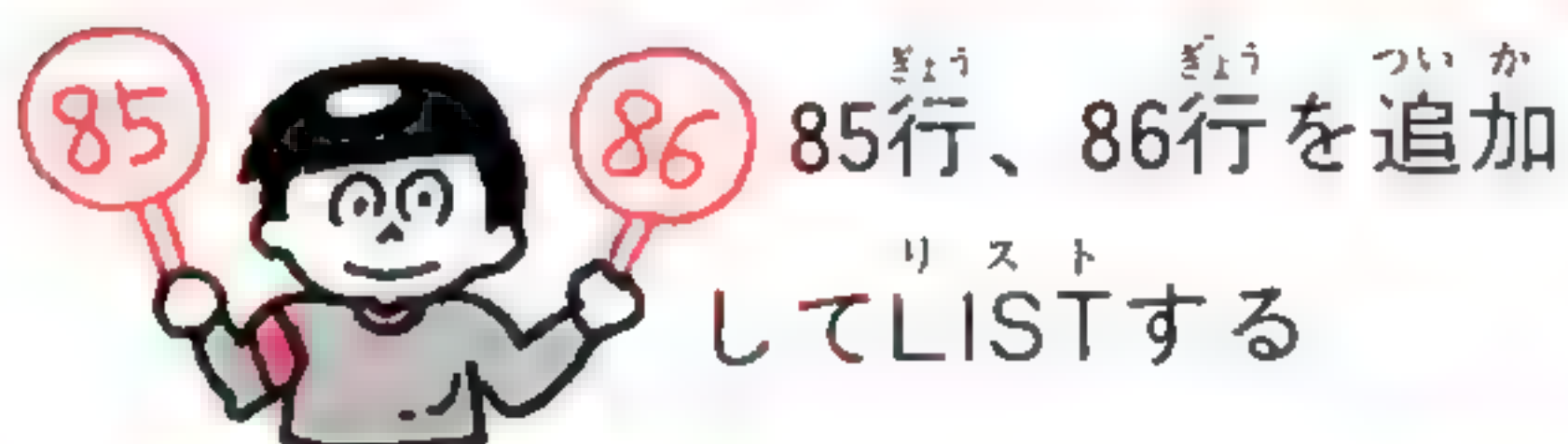
DELETE	行番号………この行番号だけを消す
DELETE	行番号①ー行番号②………①から②までの行を消す
DELETE	行番号ー………この行番号からあとの行を消す
DELETE	ー行番号………最初の行からこの行番号までを消す

●プログラムの編集をするRENUMとDELETE

①
```
60 FOR N=1 TO10
70 X=INT(RND(-TIME)*9
80 Y=INT(RND(-TIME)*9
85 IF X<Y THEN 70
86 IF(X/Y)<>INT(X/Y)
90 LOCATE 8,1+N*2
100 PRINT X;"*";Y;"="
110 LOCATE 17,1+N*2:I
```

②
```
60 FOR N=1 TO10
70 X=INT(RND(-TIME)*9
80 Y=INT(RND(-TIME)*9
90 IF X<Y THEN 70
100 IF(X/Y)<>INT(X/Y)
110 LOCATE 8,1+N*2
120 PRINT X;"*";Y;"="
130 LOCATE 17,1+N*2:I
```

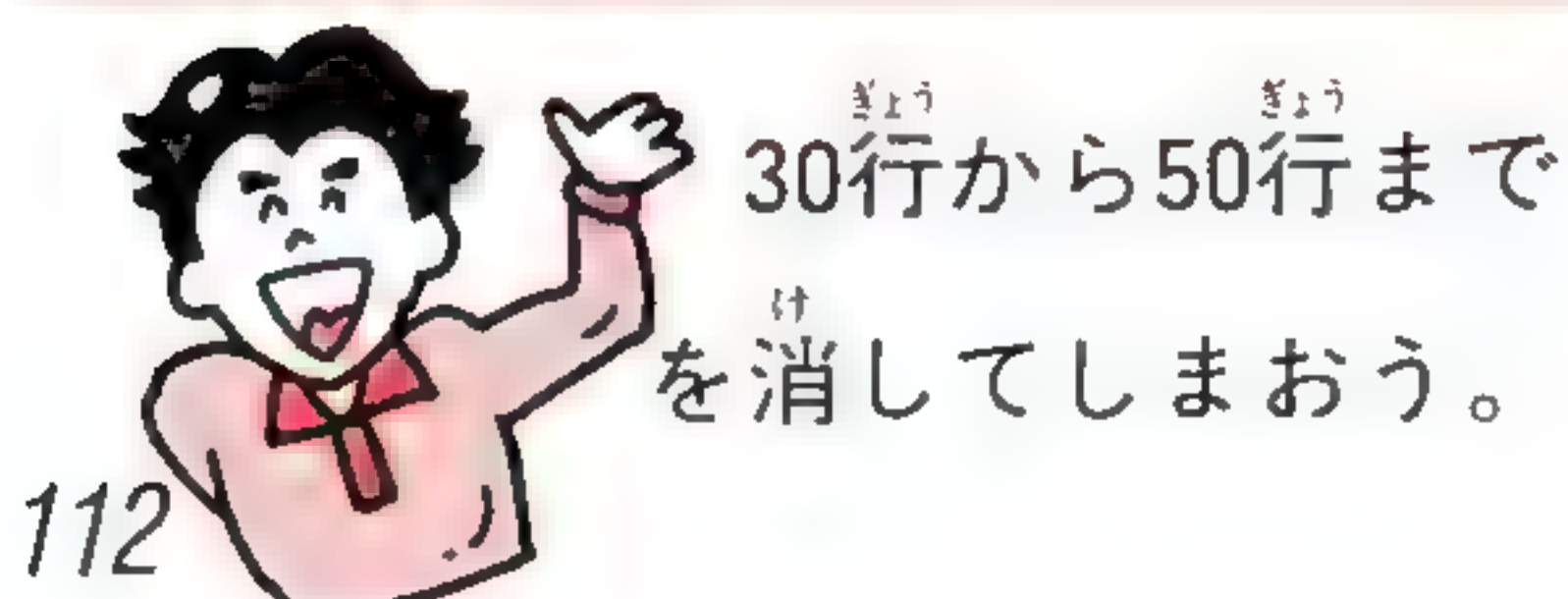
85行、86行を追加してLISTする

RENUM命令を実行すると………

③
```
10 LOCATE 10,10
20 PRINT"ABCDEF"
30 LOCATE 4,10
40 PRINT"*****"
50 PRINT"GHI"
60 END
```

④
```
10 LOCATE 10,10
20 PRINT"ABCDEF"
60 END
```

30行から50行までを消してしまおう。

DELETE　30ー50

[RETURN]キーだ。

パート4

図や絵を描いたり音を出したり

グラフィックの画面作りでスタート

SCREEN2と3は絵を描くモード

SCREEN命令●どんな画面にするかを決める

パソコンに図や絵を描かせて、色を塗ったりしてみよう。図や絵を描く場所は、もちろんディスプレイだ。でも、電源をONしたとき、ディスプレイの画面は、字を書くための画面になっているから、まず、絵が描ける画面作りからスタートだ。

画面作りの命令は、SCREENだ。

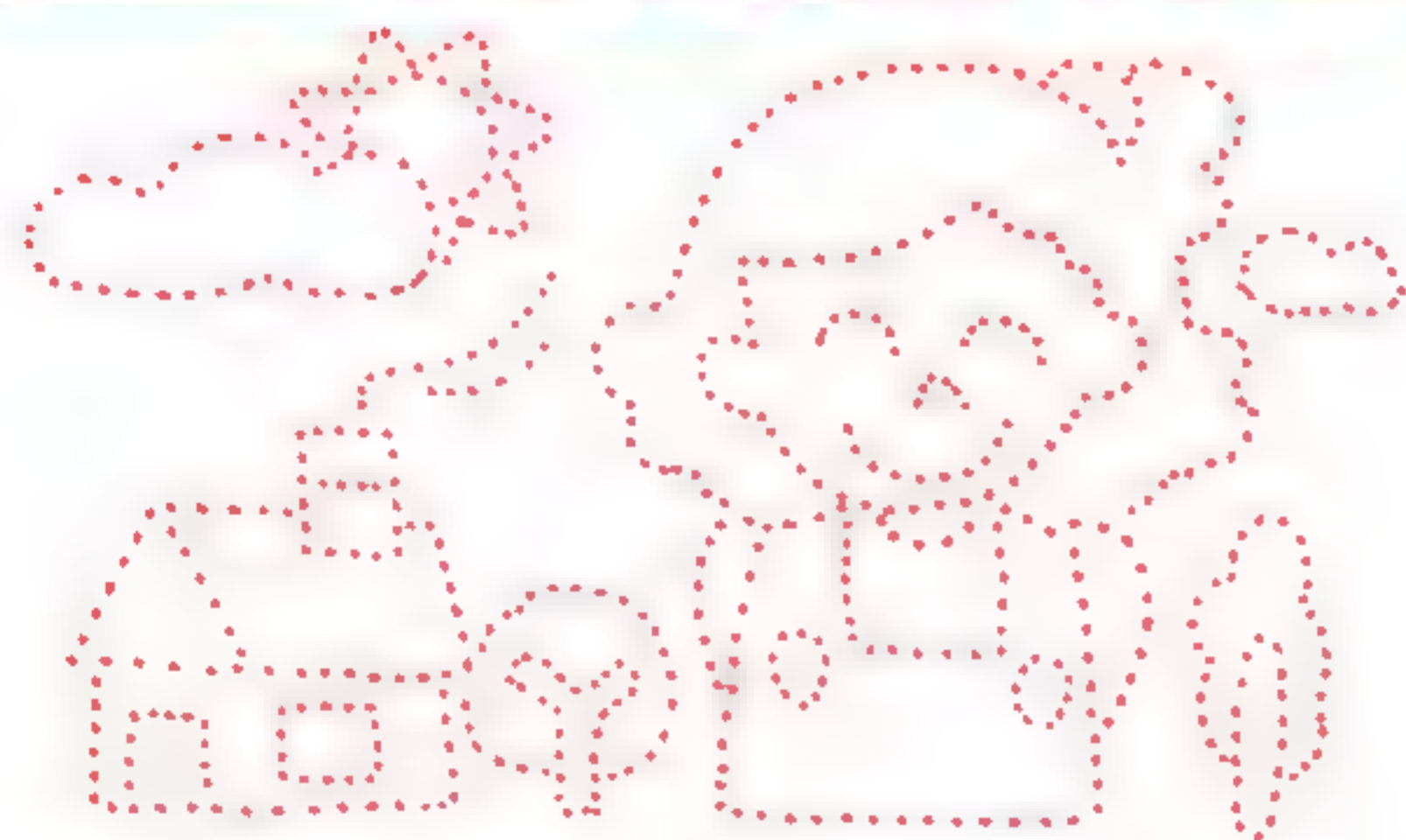
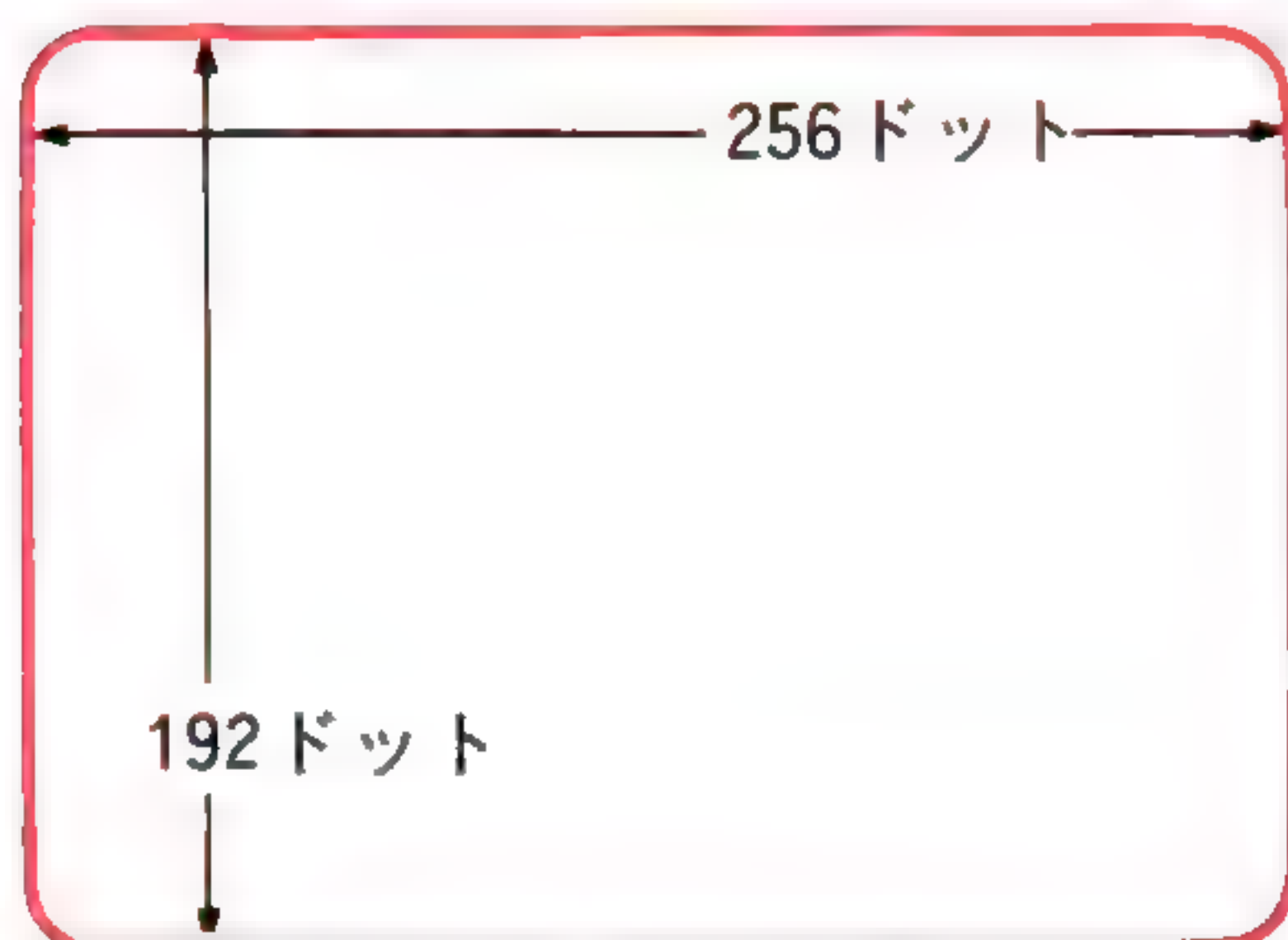
SCREEN　画面モード

という形で使う。「画面モード」というところには0～3までの数字を入れる。0と1はテキストモードだから、絵を描くなら、2か3だ。SCREEN 2で画面を作ってみよう。

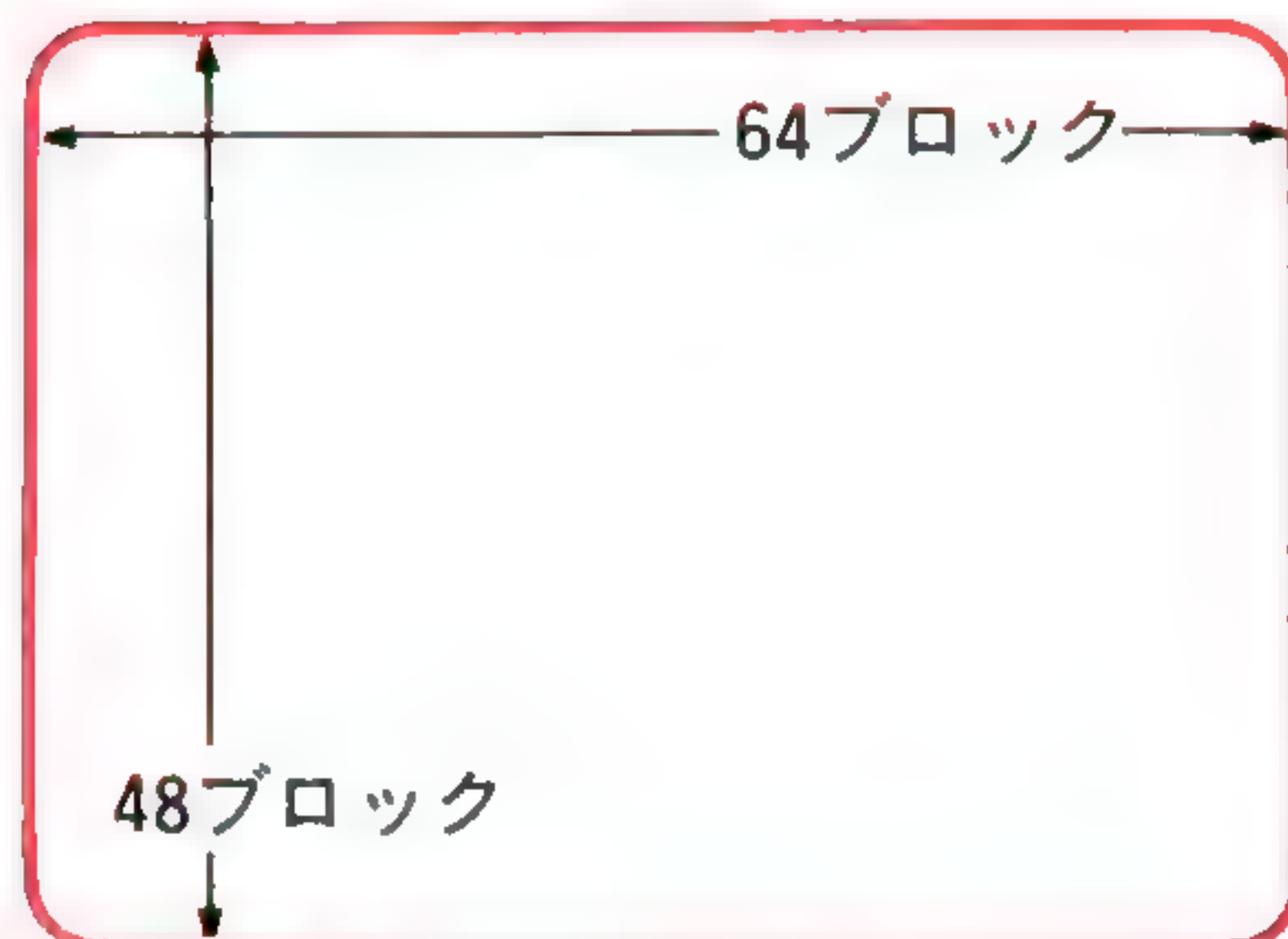
パソコングラフィックスというんだ

```
10 SCREEN 2
20 PSET(130,100),15
30 GOTO 30
```

このかんたんなプログラムを入力して実行してみよう。画面のまん中あたりに、小さな小さな白い点が表示されたはずだ。20行は、(130, 100) の位置に白い色(15)で点を描けということだよ。

●
白い点

●グラフィック画面

●SCREEN　2

高解像度グラフィックモード。細かい絵を描くことができるが、色の使い方に制約がある。

●SCREEN　3

マルチカラーモード。1ブロックの大きさは、SCREEN　2にくらべて縦横4倍だから、大まかな絵になる。

SCREEN命令

SCREEN命令で指定できる全部の要素を指定すると、次のような文型になる。

SCREEN　モード，スプライトサイズ，キークリックスイッチ，カセットボーレート，オプション

スプライトサイズ：0～3でスプライトの大きさ／キークリックスイッチ：0は音を出さない、1は音を出す／カセットボーレート：カセットに書き込む速度。1は1200ボー、2は2400ボー／オプション：0はMSX標準プリンタ、1は標準以外のプリンタ

図や絵に色をつける

使用できる絵の具は15色

COLOR命令●前景色、背景色、周辺色を決める

グラフィック画面を作って、図や絵を描いても、白と黒のモノクロじゃおもしろくない。絵の具の色は15色あるのだから、自由自在に色をつけよう。

色をつける命令はCOLOR文だ。

COLOR　前景色，背景色，周辺色

という形で使う。前景色は表示した絵の点や線につける色。背景色は画面の地の色。周辺色は画面の中で文字や形を表示できない外側のふちどりの色だ。

前景色、背景色、周辺色とも、カラーコードにある数字で指定すればいい。

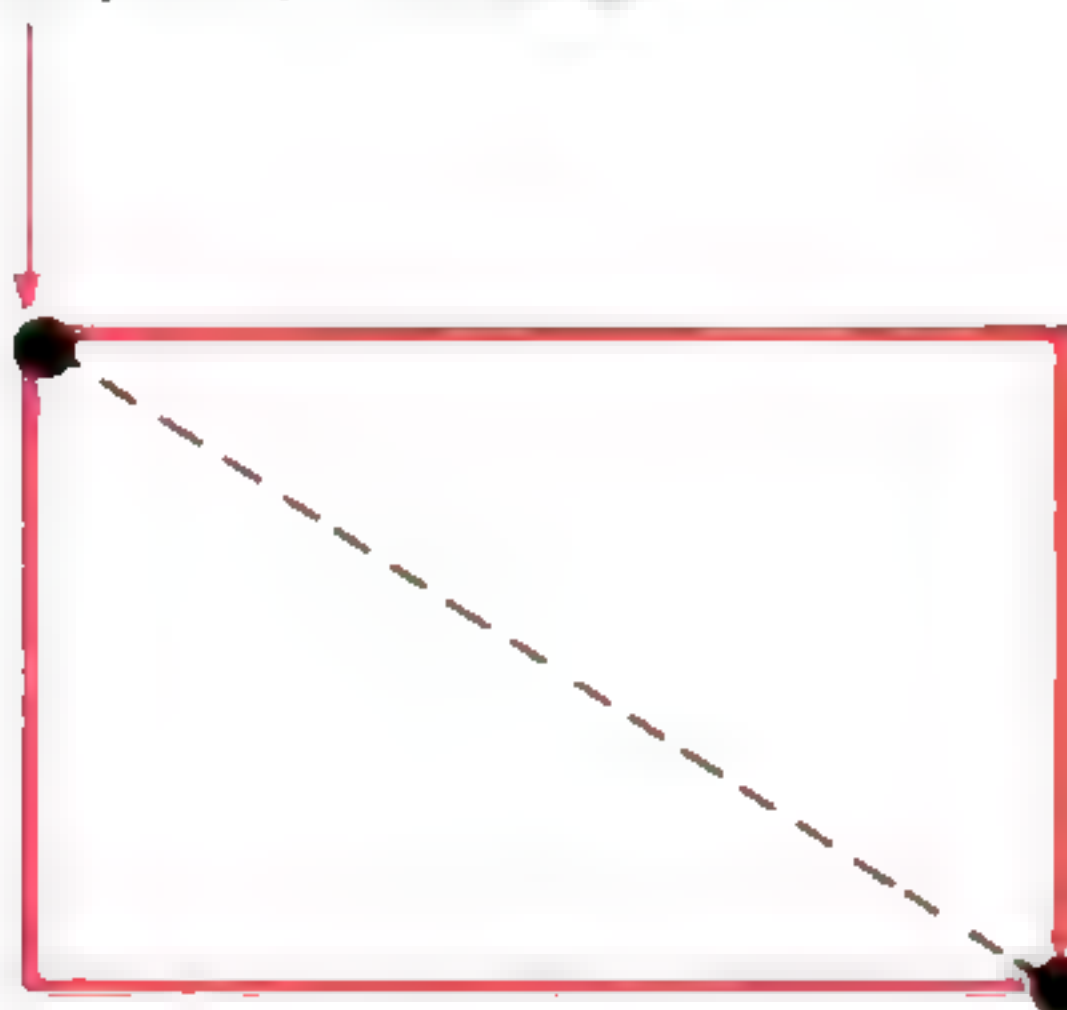
```
10 CLS
20 COLOR 13,15,1
30 SCREEN 2
40 LINE(20,20)-(150,100),,BF
50 GOTO 50
```

上のプログラムを入力して実行してみよう。紫で塗られた四角形が表示され、地の色は白で、ふちどりは黒になっただろう。

コード	色
0	周辺色と同じ色
1	黒
2	緑
3	明るい緑
4	暗い青
5	明るい青
6	暗い赤
7	水色
8	赤
9	明るい赤
10	暗い黄
11	明るい黄
12	暗い緑
13	紫
14	灰色
15	白

●絵ときCOLOR命令

図形の表示は——

```
40 LINE (20, 20) − (150, 100) , , BF
```

描いた図形に色をつけるのは——

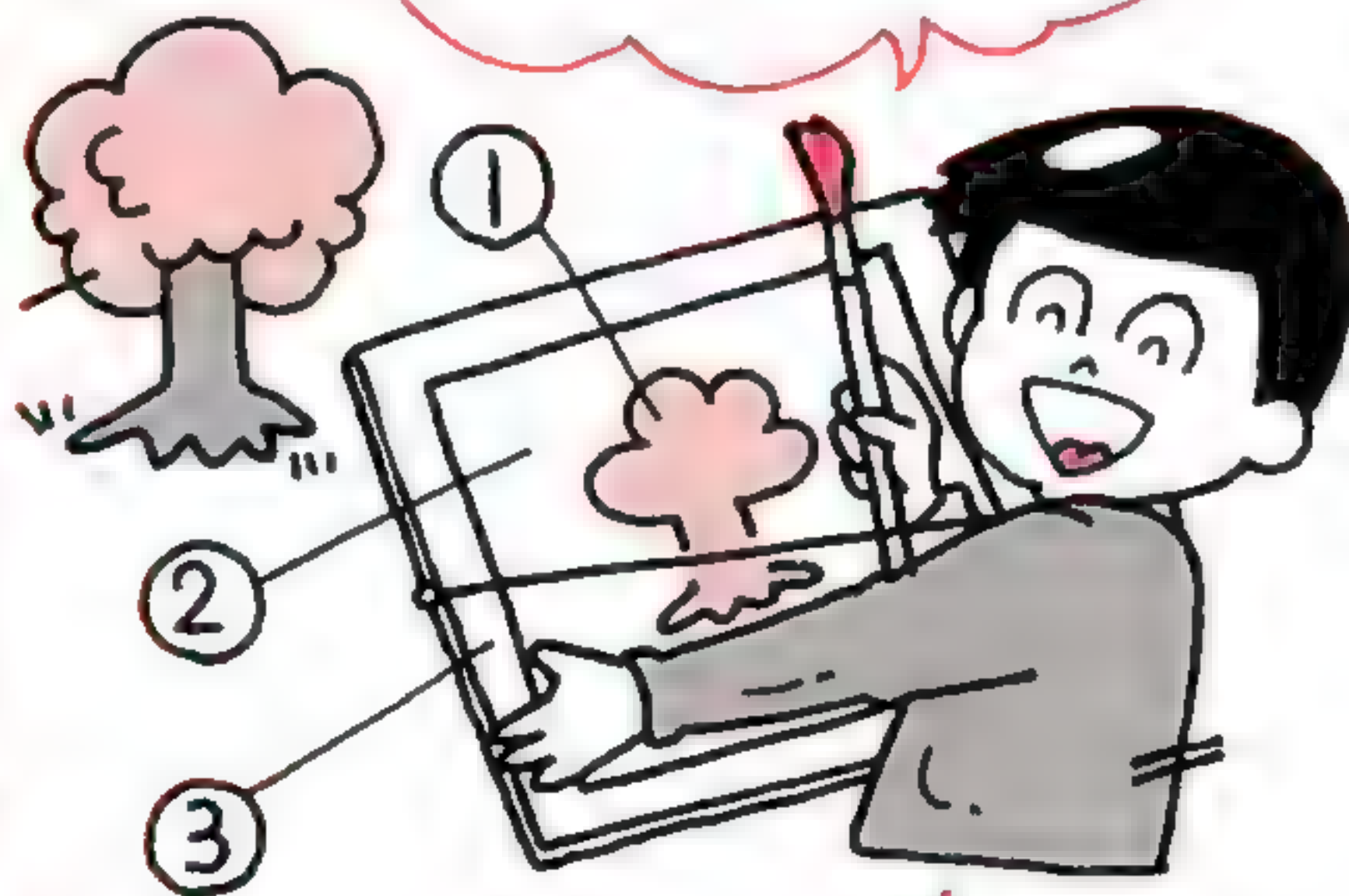
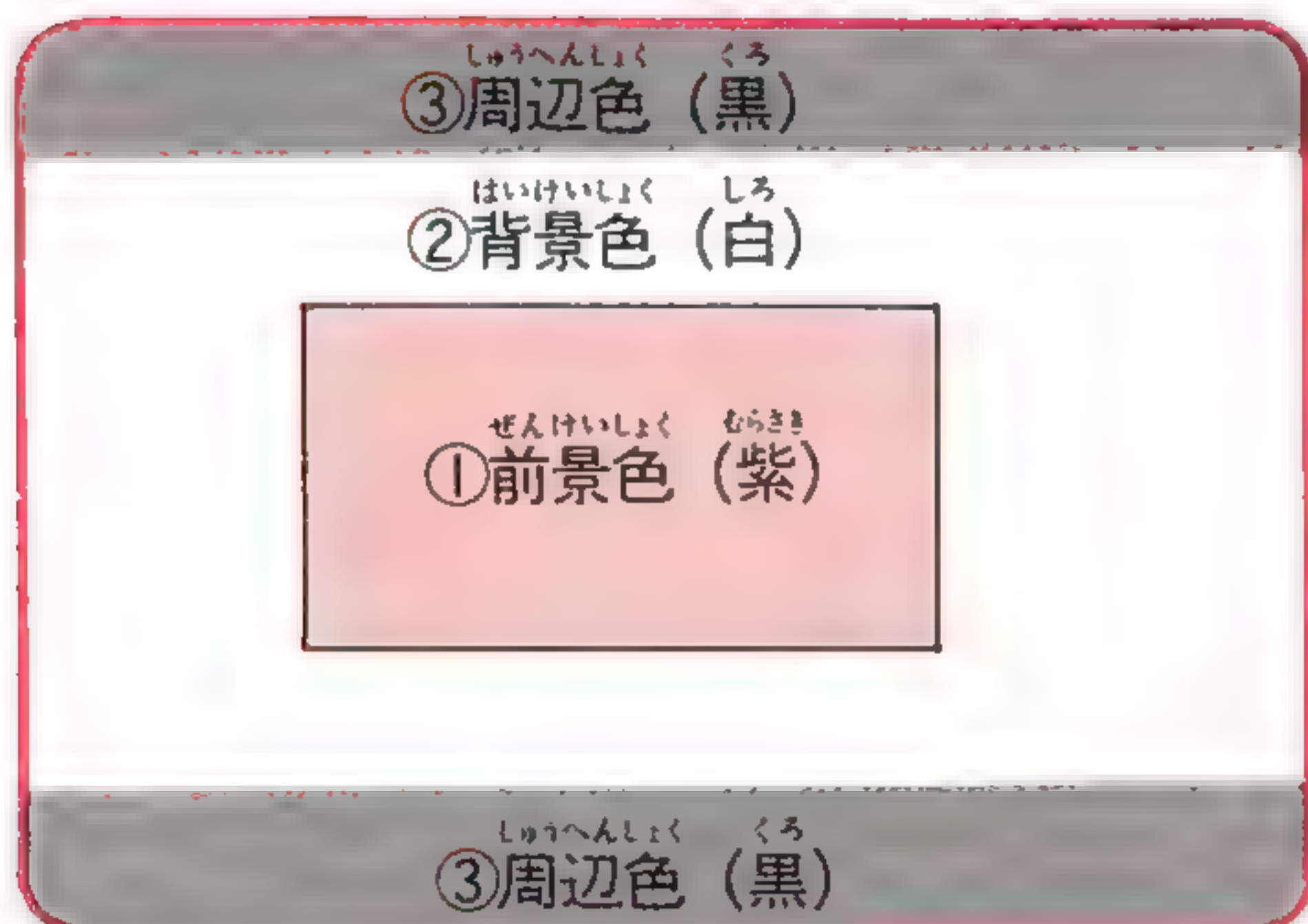
```
20 COLOR 13,15,1
```

外わくの色。1は黒だ

地の色。15は白だ

図形の色。13は紫だ

COLOR命令で指定の順はゼッタイに①前景色②背景色③周辺色の順だよ

③周辺色（黒）
②背景色（白）
①前景色（紫）
③周辺色（黒）

はじめにスイッチを入れたときは、前景色は白（15)、背景色は暗い青（4)、周辺色は水色（7）になっているよ

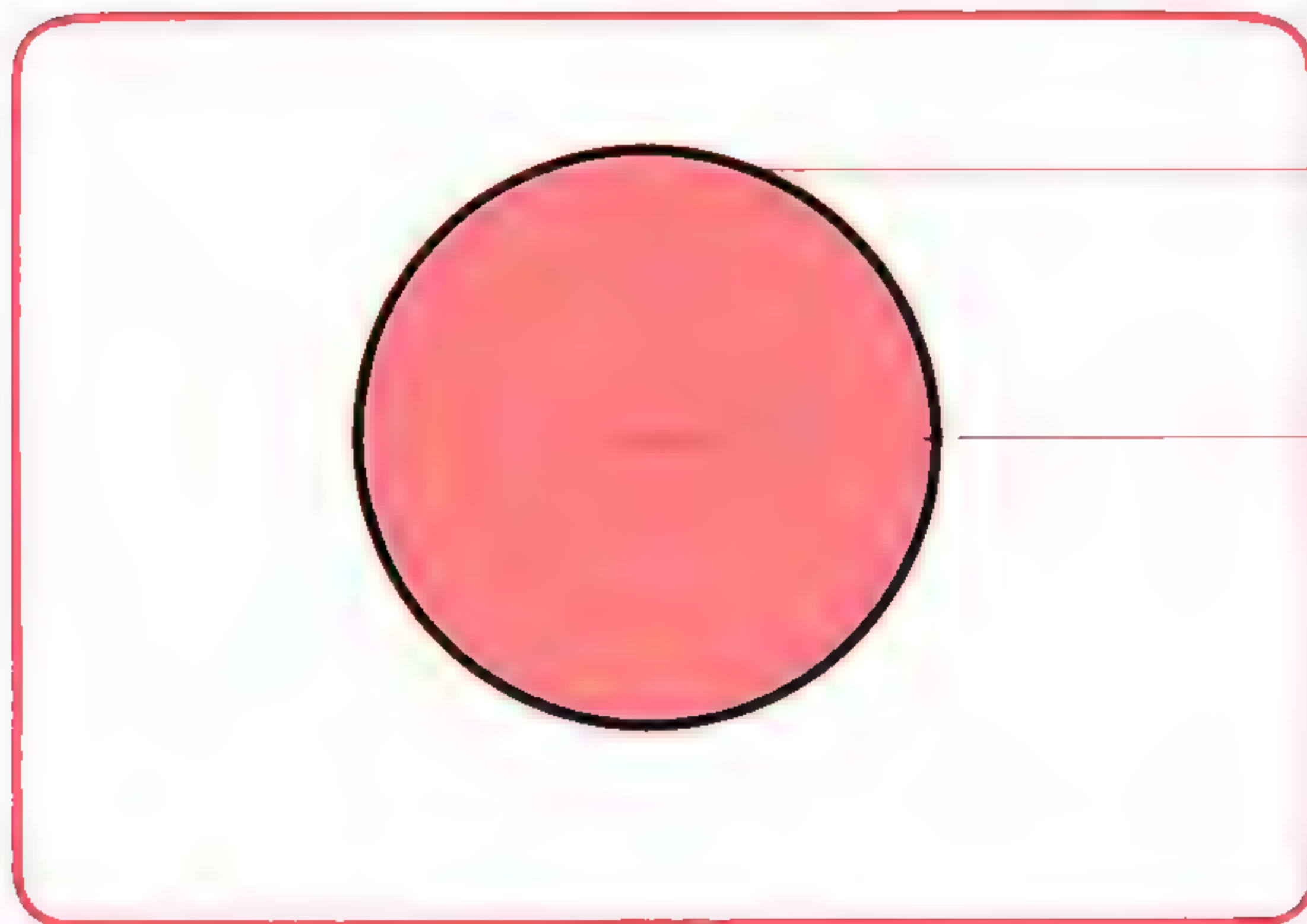
●COLOR文ショートプログラム

●カラーコードの14色をぜんぶ表示させてみる

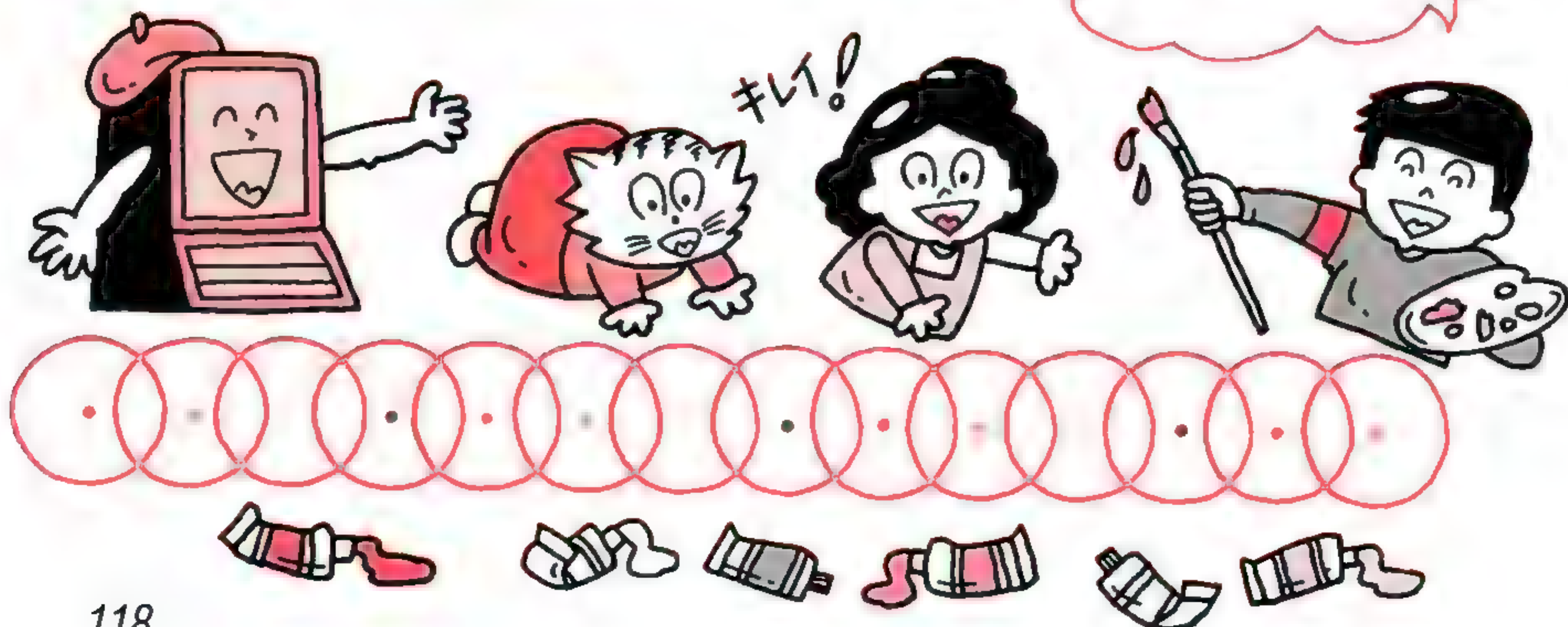
```
10 COLOR ,1
20 SCREEN 2
30 FOR I=2 TO15
40 CLS
50 COLOR I
60 CIRCLE(120,90),60
70 PAINT(120,90)
80 FOR K=1 TO500
90 NEXT K,I
100 COLOR 15,4
110 END
```

カラーコードに入っている2〜15までの色をとり出して、円の中に順番に表示させる。

60行は円を描く命令。70行のPAINT文は、そこを塗れという命令だ。

30〜90行のFOR〜NEXTで、14回色を変えている。

円を描くのはCIRCLE文。半径は60ドットだ

円の中心座標は(120,90)

円の中の色がどんどん変わるよ

直線で描くグラフィックス

まず1本の線を引いてみよう

LINE命令●線を引き、箱を描き、箱を塗る

グラフィック画面作りもやったし、色を塗る方法もすました。キャンバスと絵の具の筆が用意できたのだから、こんどは筆で、線画を描く番だ。

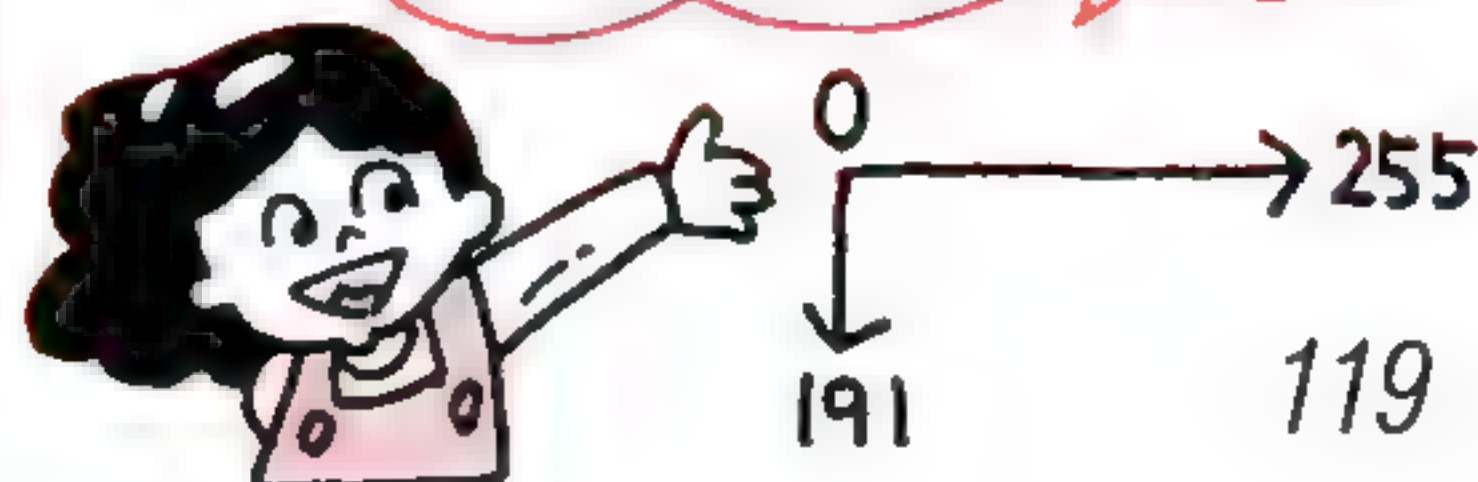
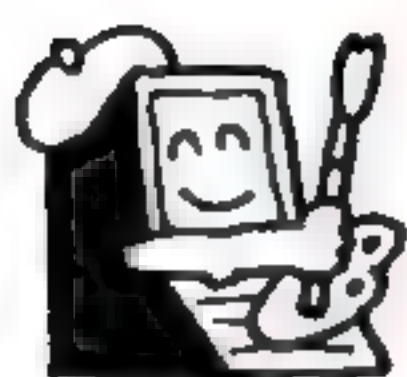
まず、線を1本引いてみよう。

```
10 SCREEN 2
20 LINE(50,50)-(150,150),9
30 GOTO 30
```

画面の横50、縦50の位置にある点と、横150、縦150の位置にある点を結んで1本の線を引け。線の色は、カラーコード9の赤だよ、という命令だ。

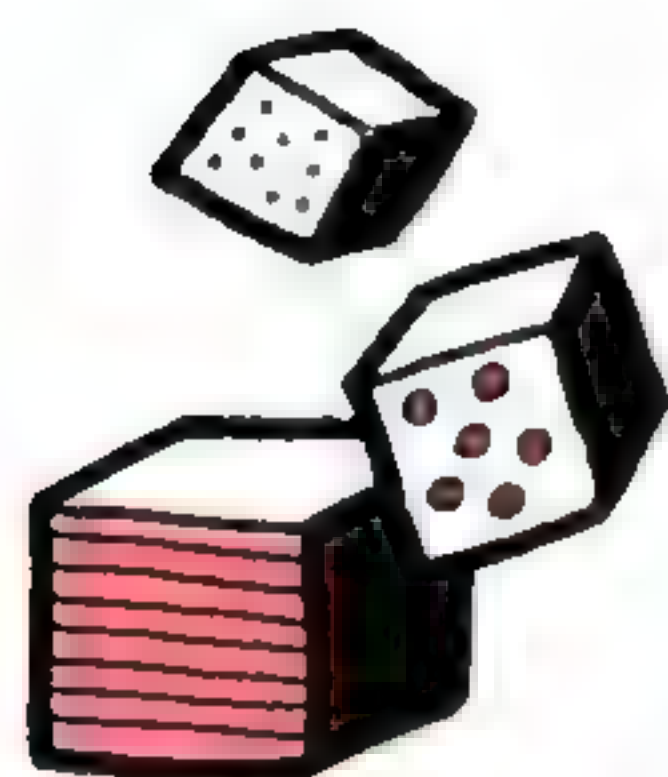
線をどこに
表示させるかは
始点の位置と
終点の位置を
決めてやればいい

横の位置はX座標
縦の位置はY座標
というんだよ

SCREEN　2の画面は
横0～255ドット
縦0～191ドットだから
その範囲で指定してね

LINE命令のフルコース

LINE命令は線を引く命令だが、ただ1本の線を引くだけではなく、1行だけのLINE命令で、四角形の箱を描いたり、描いた箱の中を色で塗ったりもできる。

LINE文は
線を引くだけじゃない

LINE　(X1，Y1) − (X2，Y2)，カラーコード，BまたはBF

- BまたはBF：Bなら箱を描き、BFなら箱を塗る
- カラーコード：線または箱の中を塗る色
- (X2，Y2)：2点間の終点の座標。Xは横、Yは縦
- (X1，Y1)：2点間の始点の座標。Xは横、Yは縦の座標

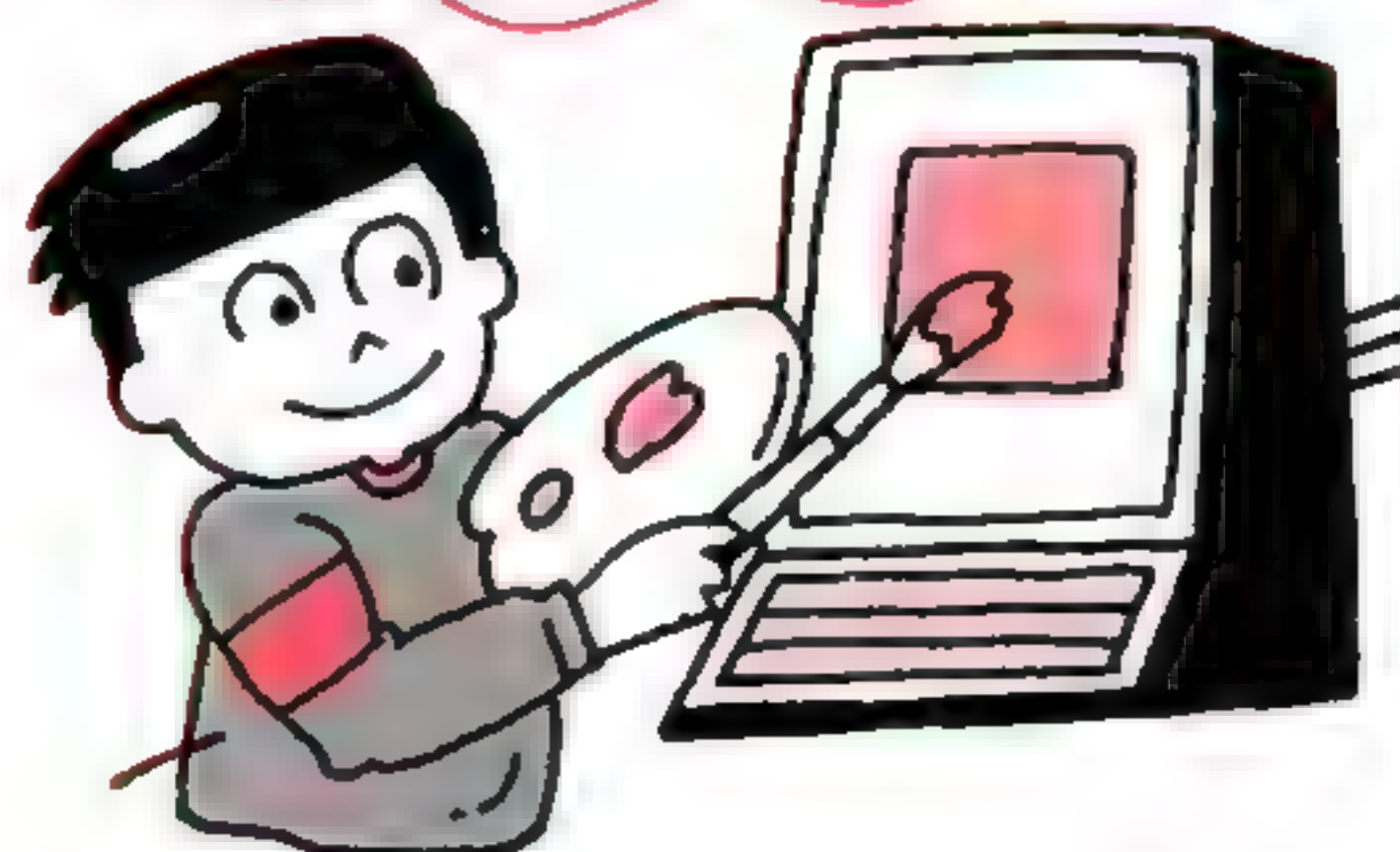
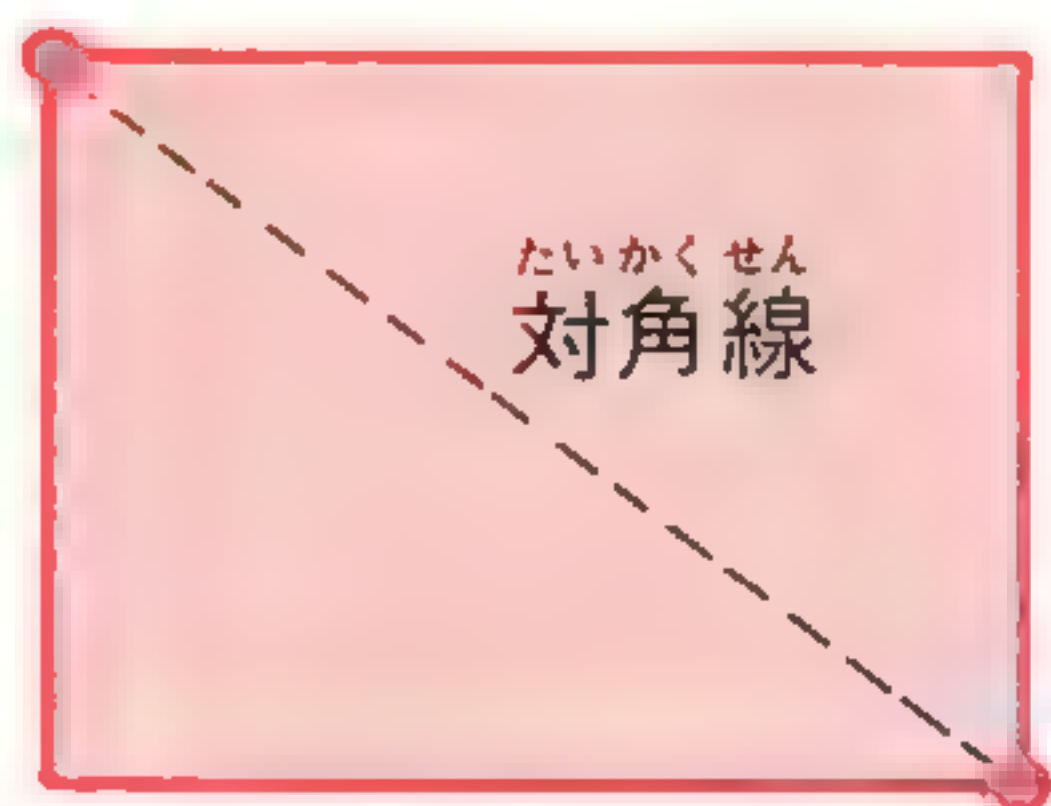
まえのページにあったLINE文にBをつけたして、実行してみよう。

```
10 SCREEN 2
20 LINE(50,50)-(150,150),9,B
30 GOTO 30
```

Bをつけたせば
指定した2点を
結んだ線を
対角線にした
四角形を描くよ

BをBFに変えれば
四角形の中を色で塗りつぶす

●LINE文ショートプログラム　1

●中心からいろいろな長さの線を描く

```
10 SCREEN 3
20 DEFINT A-Z
30 X=RND(1)*256
40 Y=RND(1)*192
50 C=RND(1)*15+1
60 LINE(127,95)-(X,Y),C
70 GOTO30
```

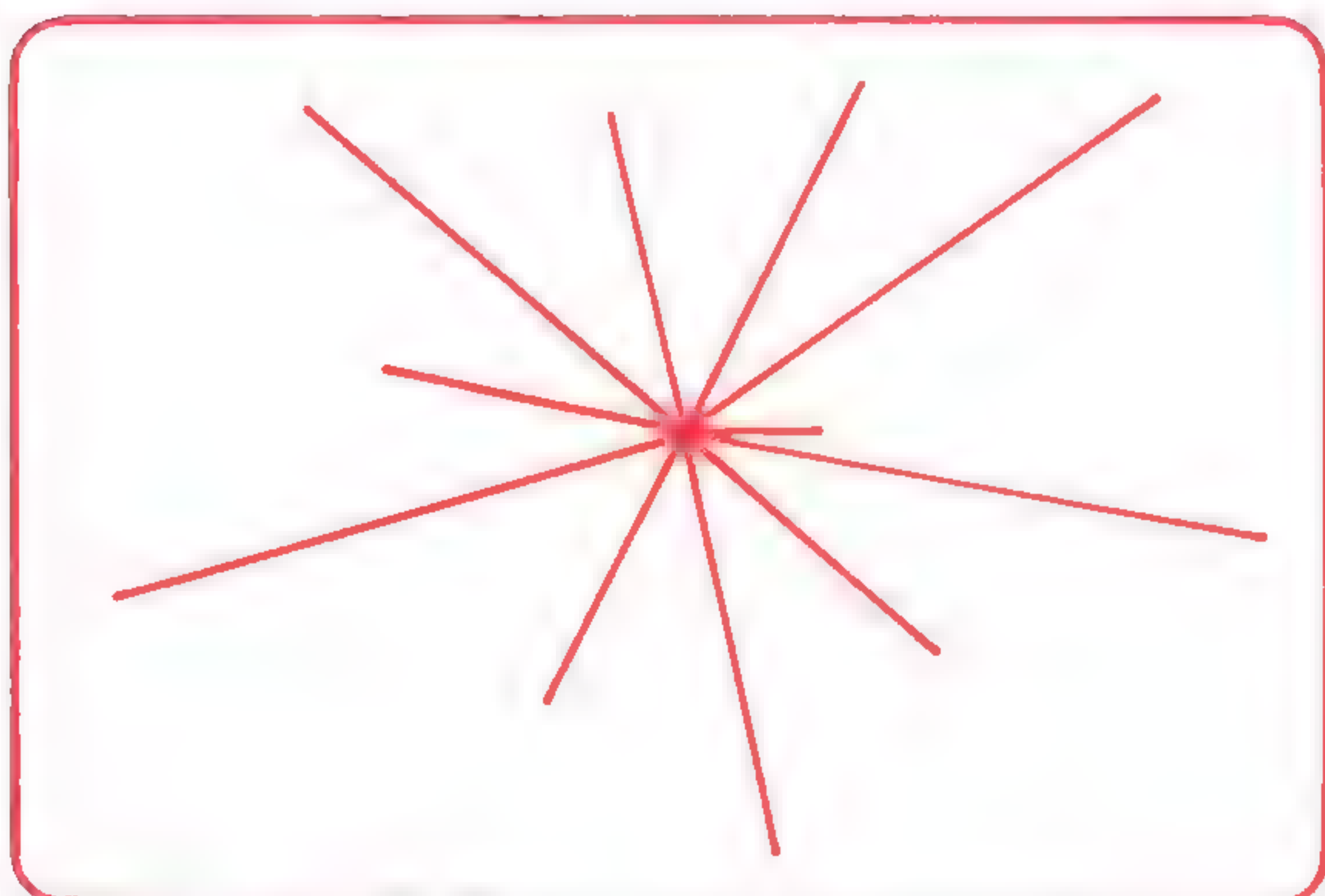
画面のほぼ中心から四方八方にいろいろな長さの線を引くプログラムだ。

SCREEN　3を使っているので、ドットが大きくなっている。

■放射状に伸びる線の基点は（127, 95）である。（直線の始点＝60行）。

■各線の終点の座標はRND関数で指定してあるからパソコンまかせだ。

■SCREEN　3の座標指定は、SCREEN　2と同じでいい。

■LINE命令で使う（X_1, Y_1）−（X_2, Y_2）の「−」は、マイナスの記号である。

(注　意) グラフィック命令のプログラムで、ENDで終わっていないプログラムを止めるには、CTRL キーと STOP キーを同時に押す。

五角形でも六角形でも描けるよ

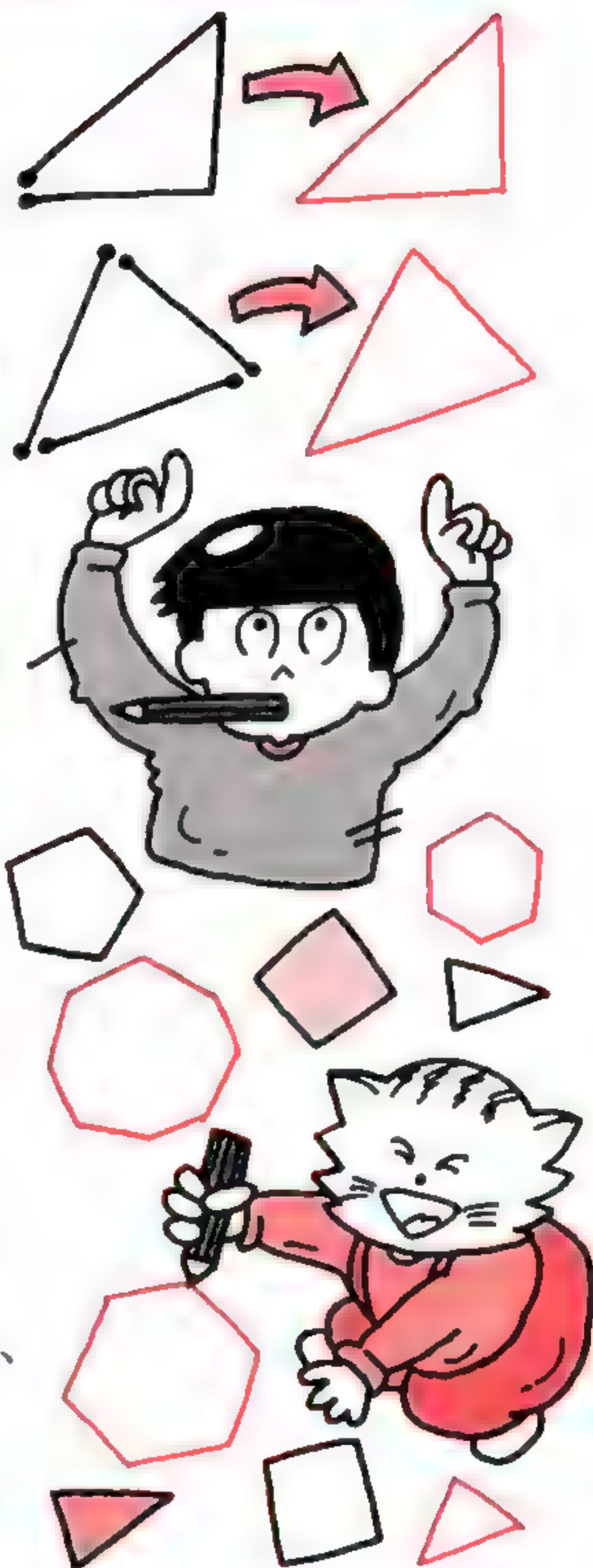
LINE文で四角形を描くのは、終わりにBをつければいい、と習ったね。もちろんBをつけないで、LINE文を4本並べても同じだ。

```
10 SCREEN 2
20 LINE(50,50)-(150,50)
30 LINE(50,50)-(50,150)
40 LINE(150,50)-(150,150)
50 LINE(50,150)-(150,150)
60 GOTO 60
```

上のプログラムの4本のLINE文は、

LINE (50, 50)−(150, 150), , B

と同じなのだ。

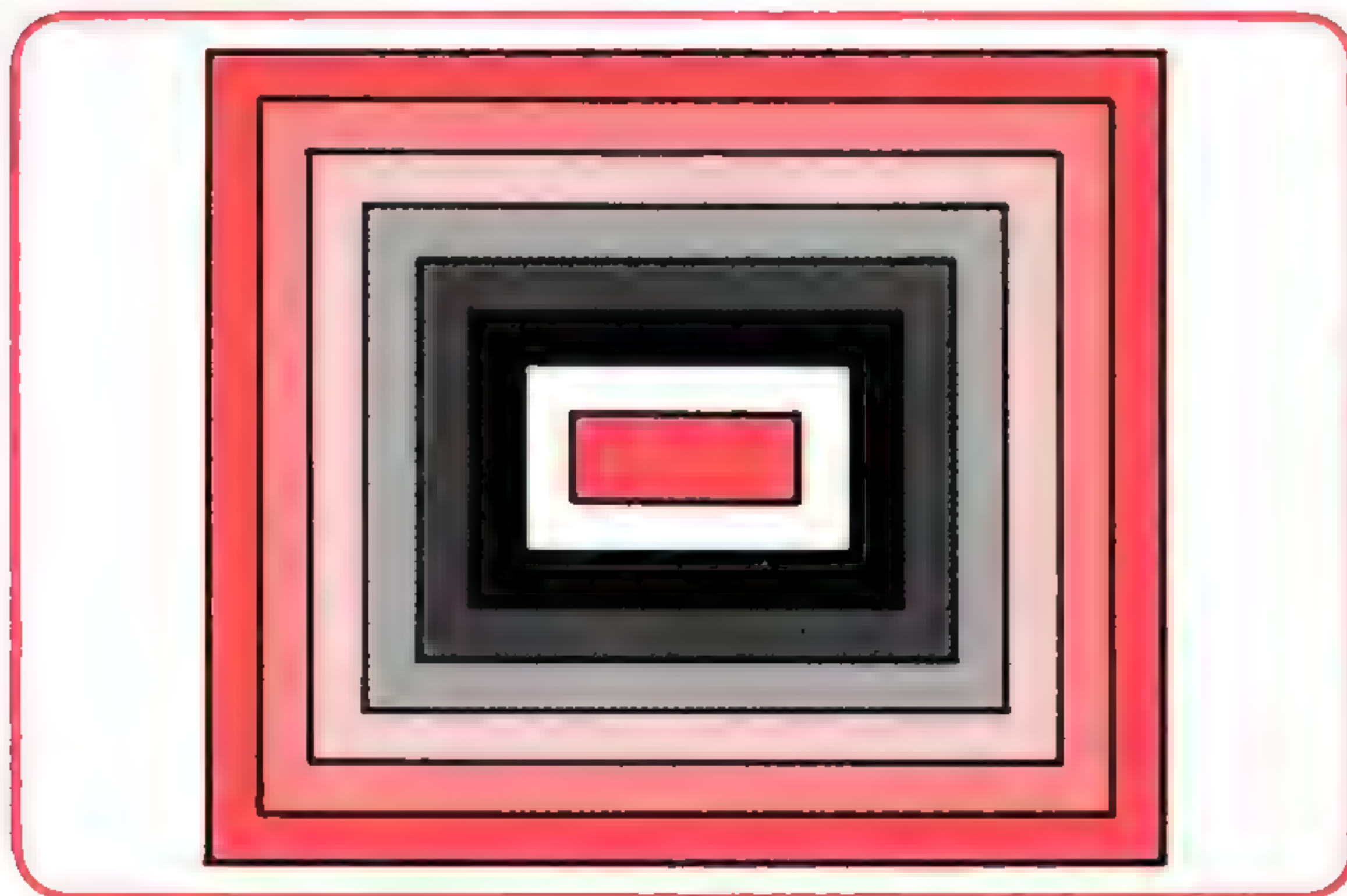
同じように考えれば、LINE文を3行書いて三角形が描けるし、5行書いて五角形が描けるのは、かんたんにわかるよね。八角形でも十角形でも、なんでも描けるはずだ。

STEPつきLINE文

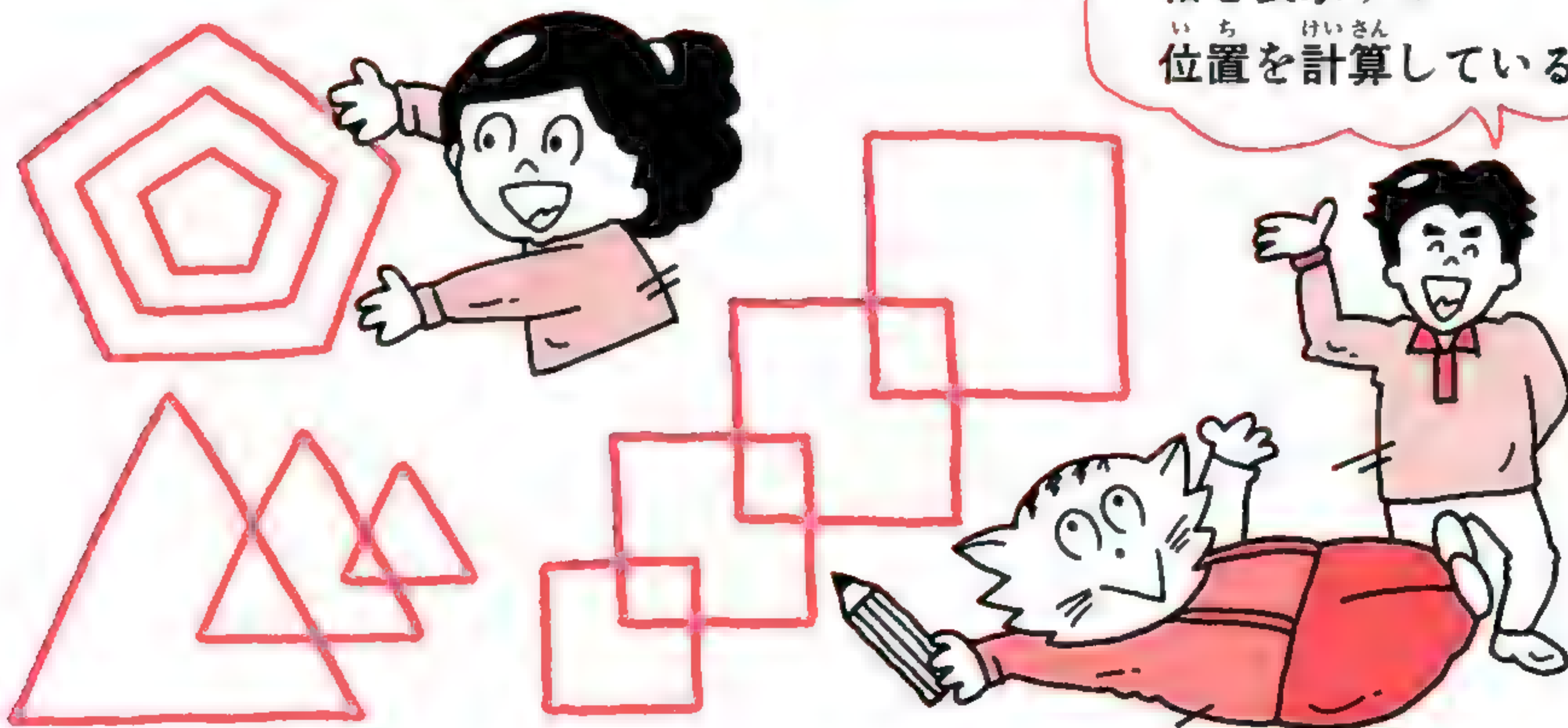
LINE文には、LINE　STEP (X_1, Y_1) −STEP (X_2, Y_2) という文型もある。STEP (X_1, Y_1) は、その直前に使った (X_1, Y_1) の座標点から、(X_1, Y_1) だけずれた座標点のことだ。たとえば、直前に使った座標が(100, 50) で、いま、STEP (10, 10) とすれば、(110, 60) の位置ということだ。

●LINE文ショートプログラム　2

●箱の中にいくつも箱を描く

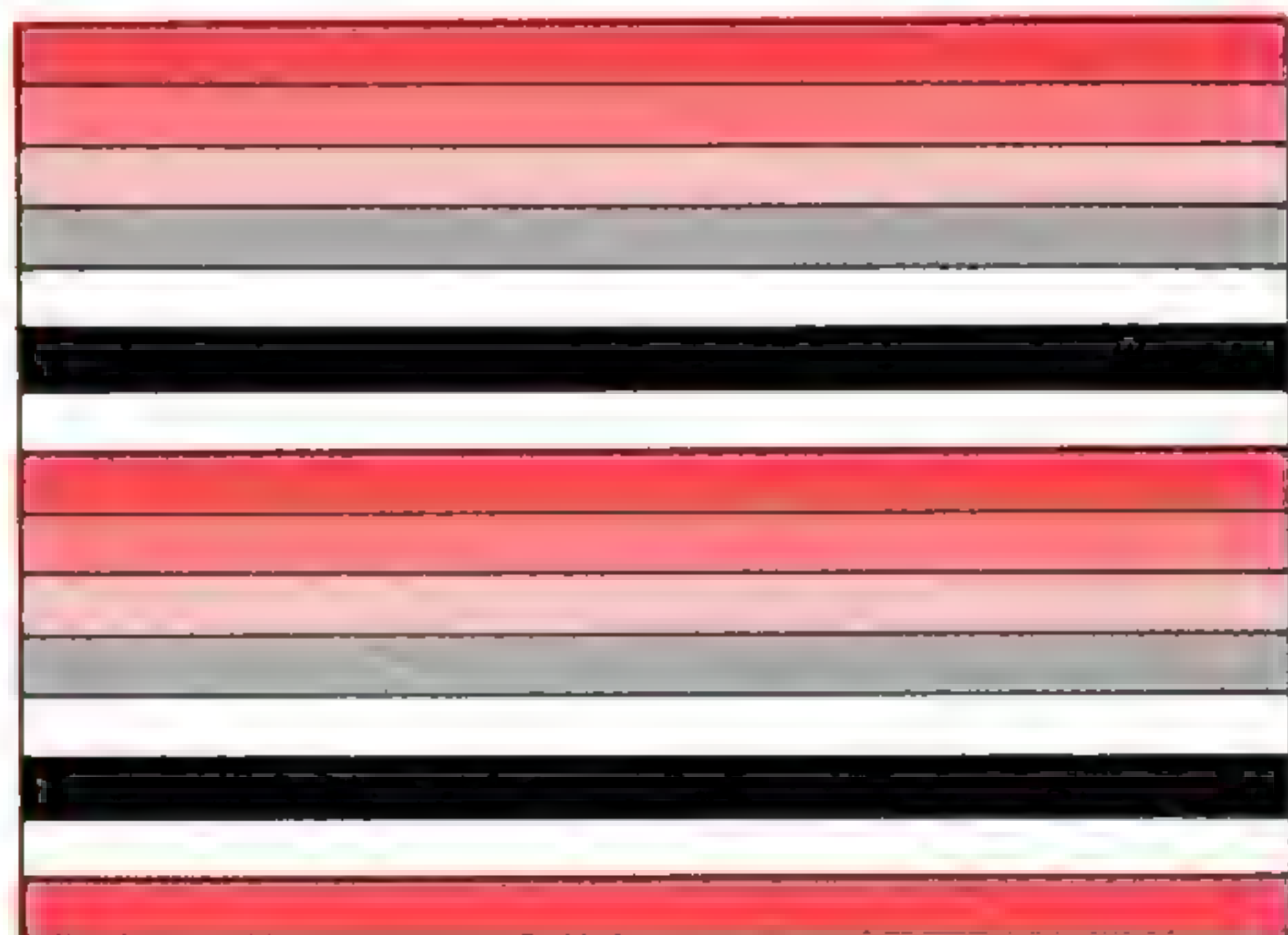
```
10 SCREEN 2
20 FOR I=1 TO15 STEP 2
30 X1=10+I*5
40 Y1=10+I*5
50 X2=240-I*5
60 Y2=180-I*5
70 LINE(X1,Y1)-(X2,Y2),I,B
80 NEXT
90 GOTO90
```

箱の中に箱を描き、またその箱の中に箱を描き、結局8個の箱を描いてしまうプログラム。箱を表示する色は、ぜんぶ違うのだ。

20行の変数Iは
LINE命令の
色のカラーコードを
変化させているよ

30行～60行は
箱を表示する
位置を計算している

●LINE文ショートプログラム　3

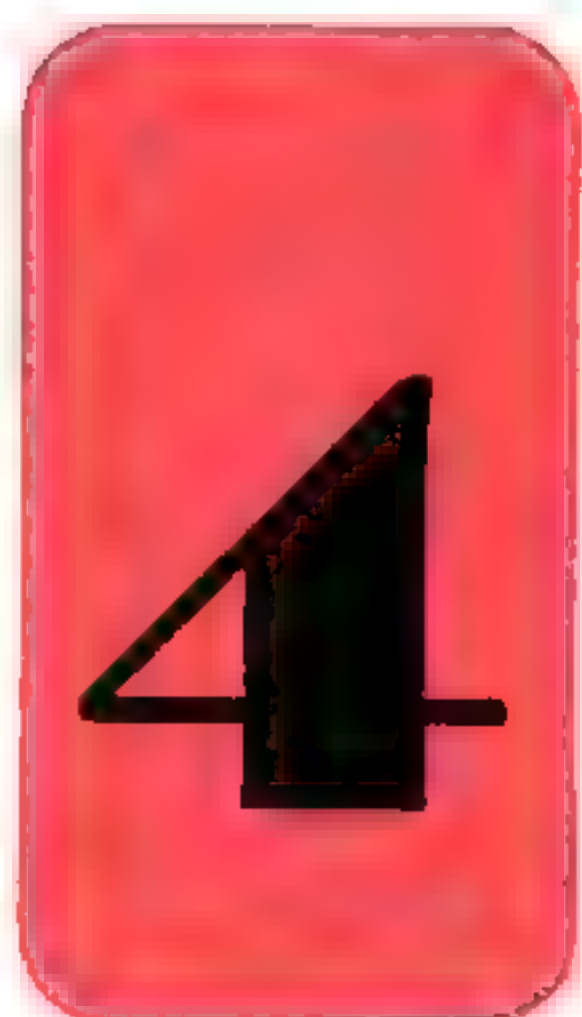
●細長い箱を15個描いてそれぞれ色を塗る

```
10 SCREEN 2
20 FOR I=1 TO15
30 Y1=(I-1)*12+5
40 Y2=Y1+11
50 LINE(10,Y1)-(240,Y2),I,BF
60 NEXT
70 GOTO70
```

横に細長い四角形を縦に15個並べて、それぞれ別の色で表示するプログラムだ。

終わりに
BFをつければ
箱を描いて
中を塗るのだったね

箱の中の色塗りは
LINE文に
BFをつけないで
PAINT命令で
塗っても同じだ
でも、このほうが
かんたんだよね

LINE文で色のコードを指定しないときは、50行のＩはいらない。ただし、Ｉの両わきのコンマ(,)は省略できない。次のようになる。

LINE $(X_1, Y_1) - (X_2, Y_2)$, , BF

円、だ円、扇形を描く

中心はどこ？半径は？どの色で？

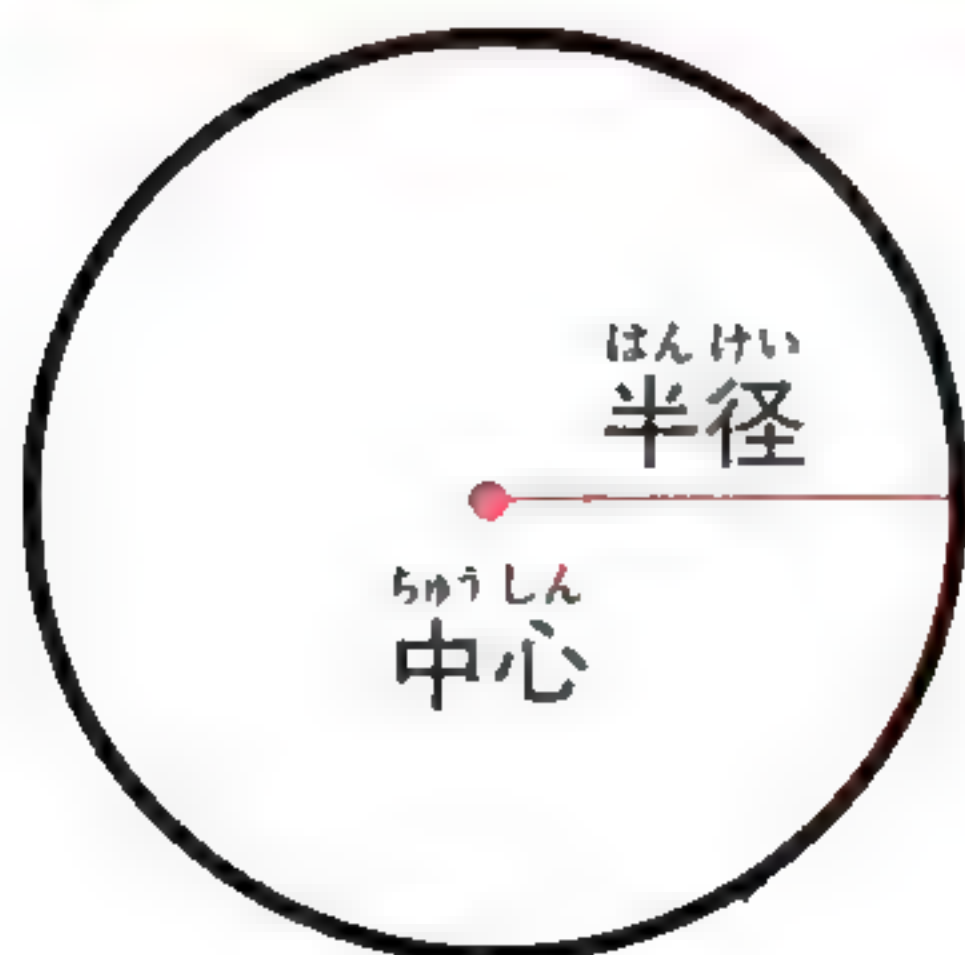
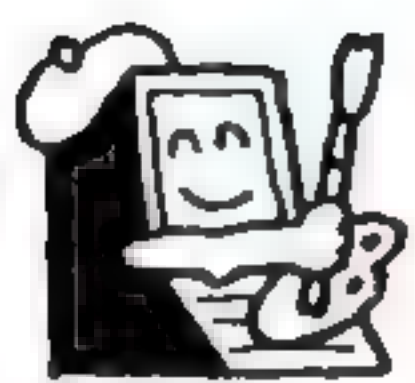
CIRCLE命令●だ円も描けるパソコンのコンパス

円やだ円、扇形を描くのはCIRCLE命令だ。
CIRCLE命令の基本形はかんたんだ。

ＣＩＲＣＬＥ　（Ｘ，Ｙ），半径，カラーコード

と、これでいい。

コンパスで円を描くときだって、中心の位置を決めてから、コンパスを半径の長さに開いて、グルッとまわすだろう。あれと同じだよ。
（X，Y）は中心の横の位置と縦の位置だ。半径はドットの数だ。そして、その円をどんな色で描くかを、カラーコードで指定してやればいい。

```
10 SCREEN 2
20 CIRCLE(130,90),50,8
30 GOTO 30
```

開始角度と終了角度を加えれば円弧

CIRCLE文はコンパスだから、ぐるりとまわって円を描いてしまう。その円周上を、ここからここまでしか描かなくていいよ、と命令できれば、円弧（アーチ）ができるはずだ。

円を描く基本形を思い出してみよう。

CIRCLE　(X，Y)，半径，カラーコード

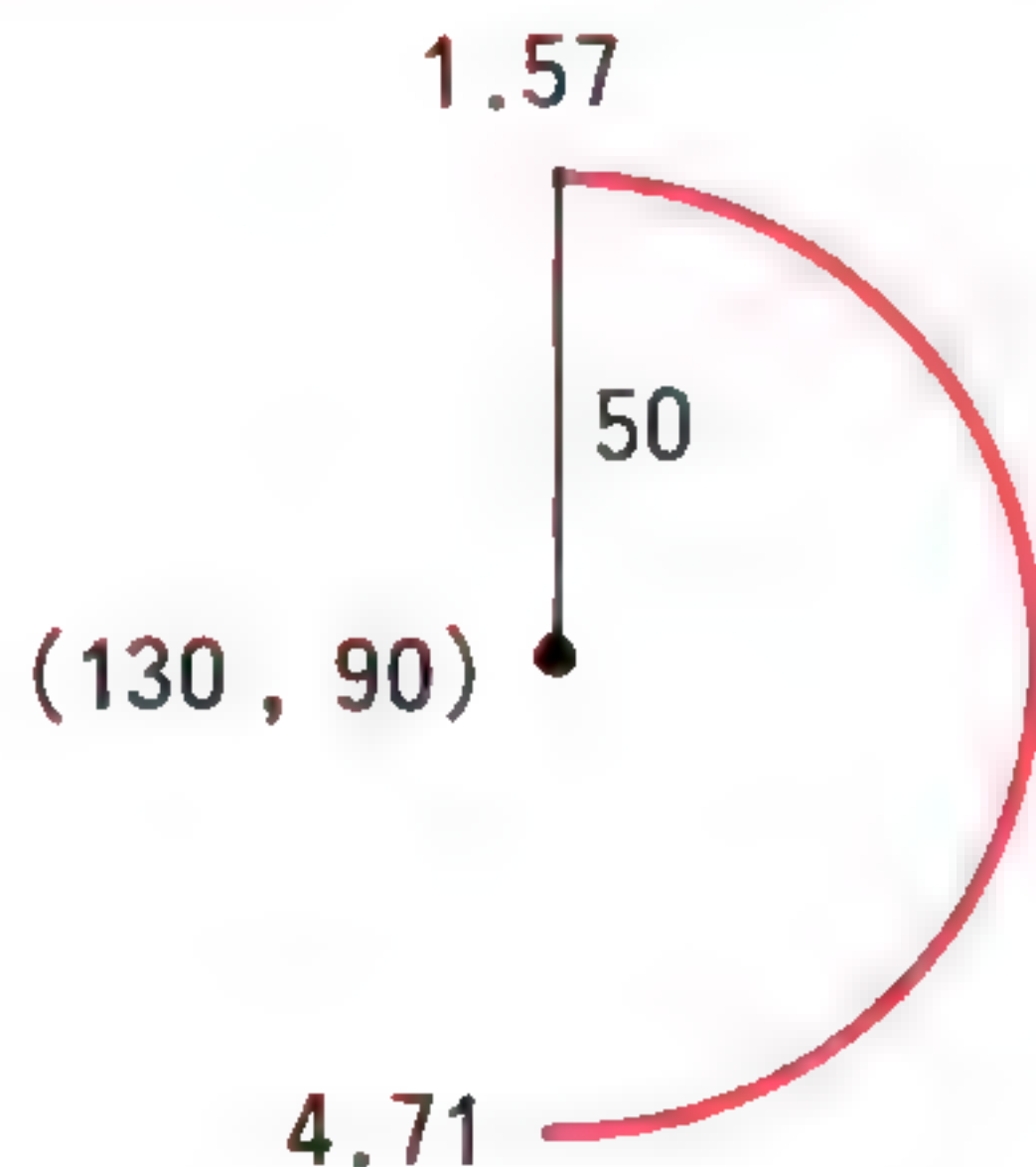
だったけれど、これに「開始角度」と「終了角度」というものを加えてみる。

CIRCLE　(X，Y)，半径，カラーコード，開始角度，終了角度

ということになる。

これが円弧を描くCIRCLE文だ。

```
10 SCREEN 2
20 CIRCLE(130,90),50,8,4.71,
   1.57
30 GOTO 30
```

画面にはどんな図が描かれたかな？

角度はラジアンで
指定するのだが
むずかしいから
ほかの方法で
おぼえたほうが
かんたんだよ

ラジアンの計算

ラジアンを正確に計算する公式は

角度×π／180

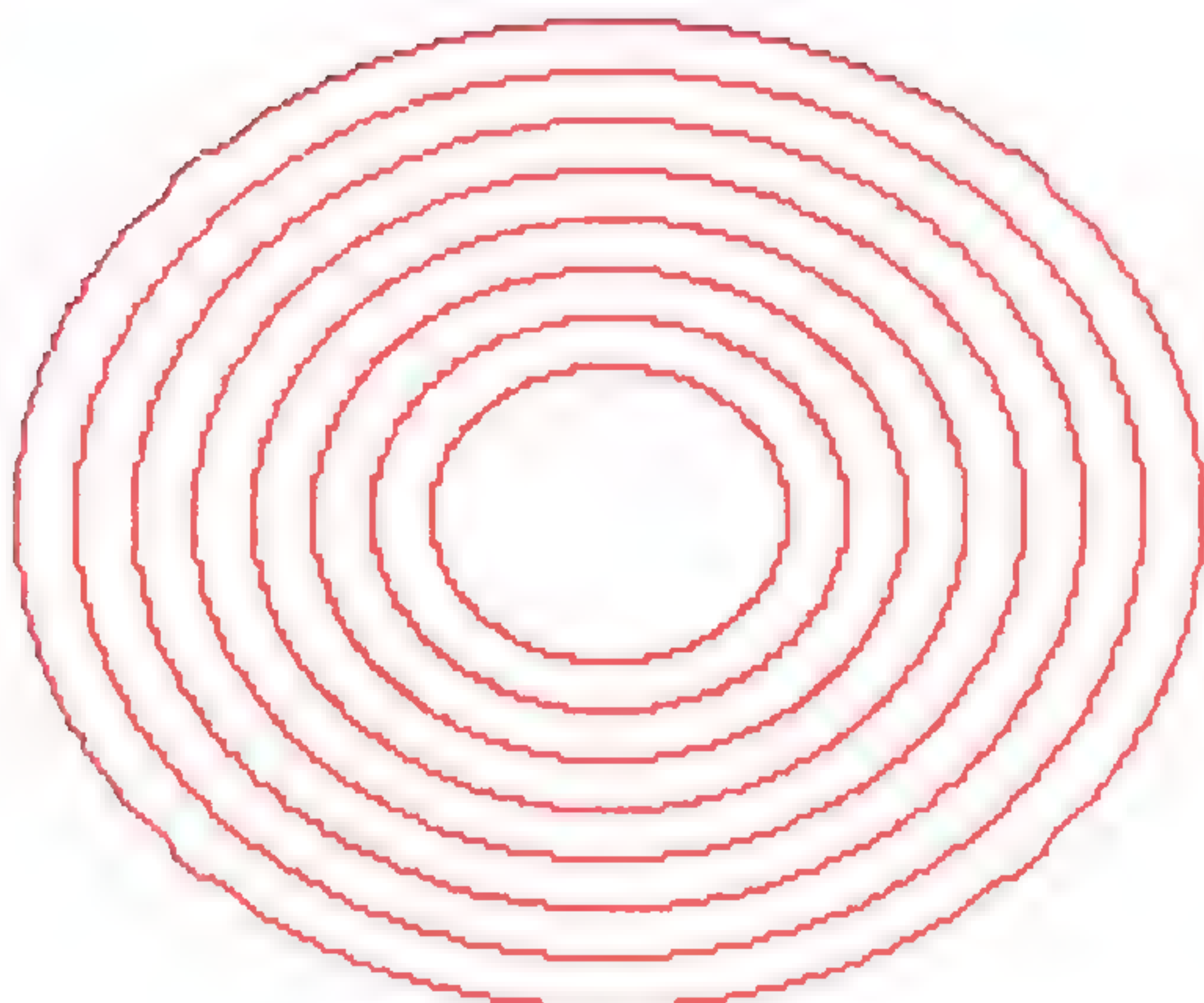
だ。0°から180°の半円を描く終了角度は、

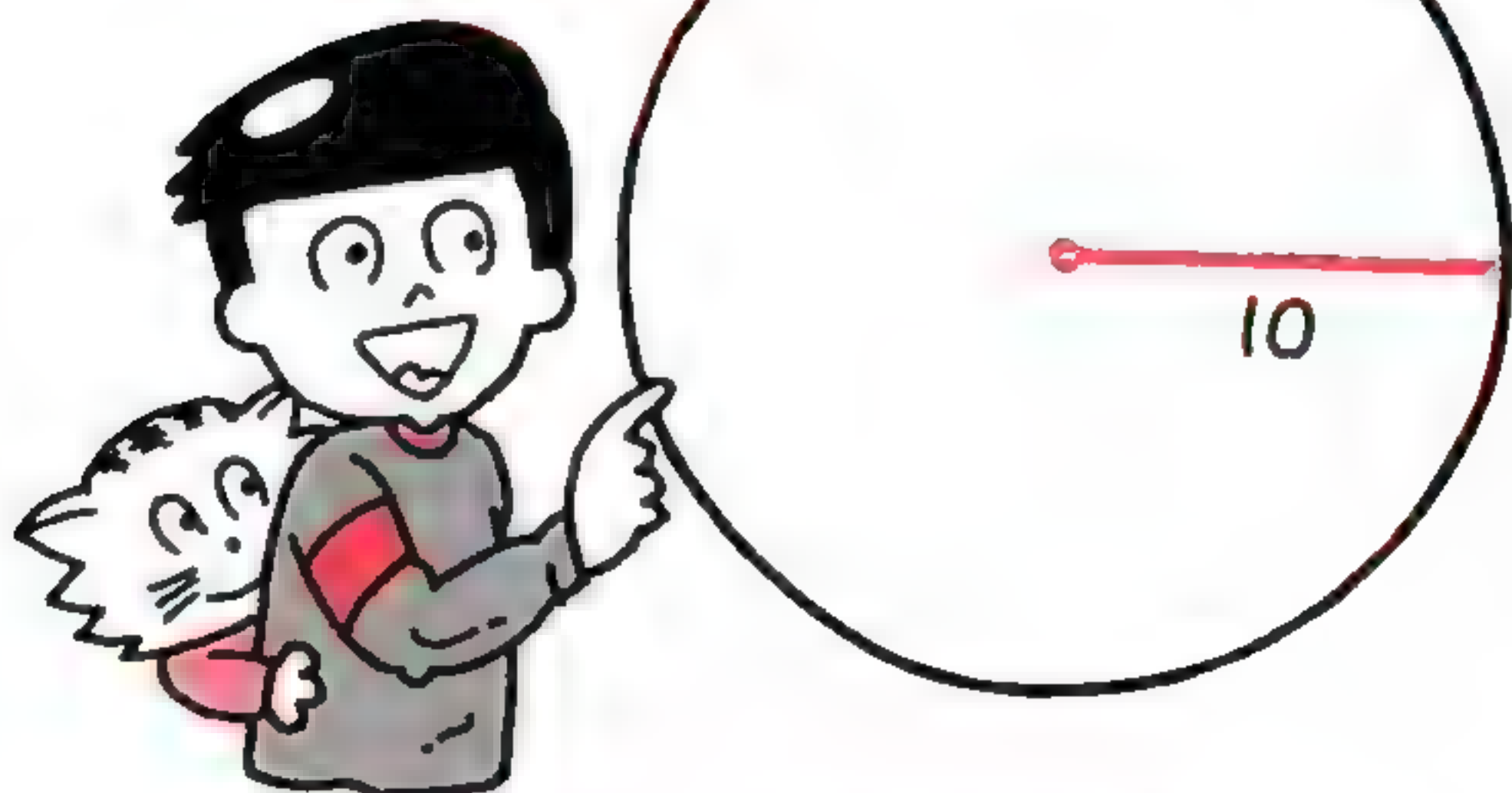
180°×3.14÷180
＝3.14ラジアン

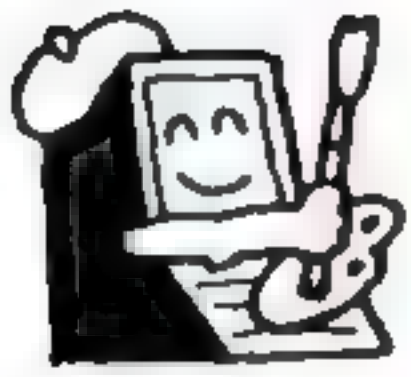
180°が3.14だから、90°なら半分の1.57というように、だいたいの数を暗算してもいい。

●CIRCLE文ショートプログラム　1

●画面に8個の円を8色使って描く

```
10 SCREEN 2
20 FOR I=1 TO15 STEP2
30 R=I*4+20
40 CIRCLE(120,90),R,I
50 NEXT
60 GOTO 60
```

20行のFOR文は、描く色の指定をしている。カラーコード1～15までの色を、STEP 2で順に描いていく。

だから、カラーコード1、3、5、7、9、11、13、15の8色だ。

40行のCIRCLE文で、Rは半径だ。30行で計算した8通りの値が半径になる。

なぜ8通りかというと、もう一度20行を見よう。20行のIは8通りあるから、40行のIも同じく8通りあると考えようよ。

角度にマイナスをつければ扇形だ

CIRCLE文の基本形に開始角度と終了角度をつけたして、アーチ（円弧）を描く方法をおぼえたけれど、これを応用すれば、扇形も描けるはずだ、とは考えられないかい？

アーチを描いて、その線の端と円の中心を線で結べば、扇形ができるのは、だれにもわかるよね。

もったいぶらないで、教えちゃおう。アーチを描くときの開始角度と終了角度にマイナス（－）をつければ、円弧のできあがり。

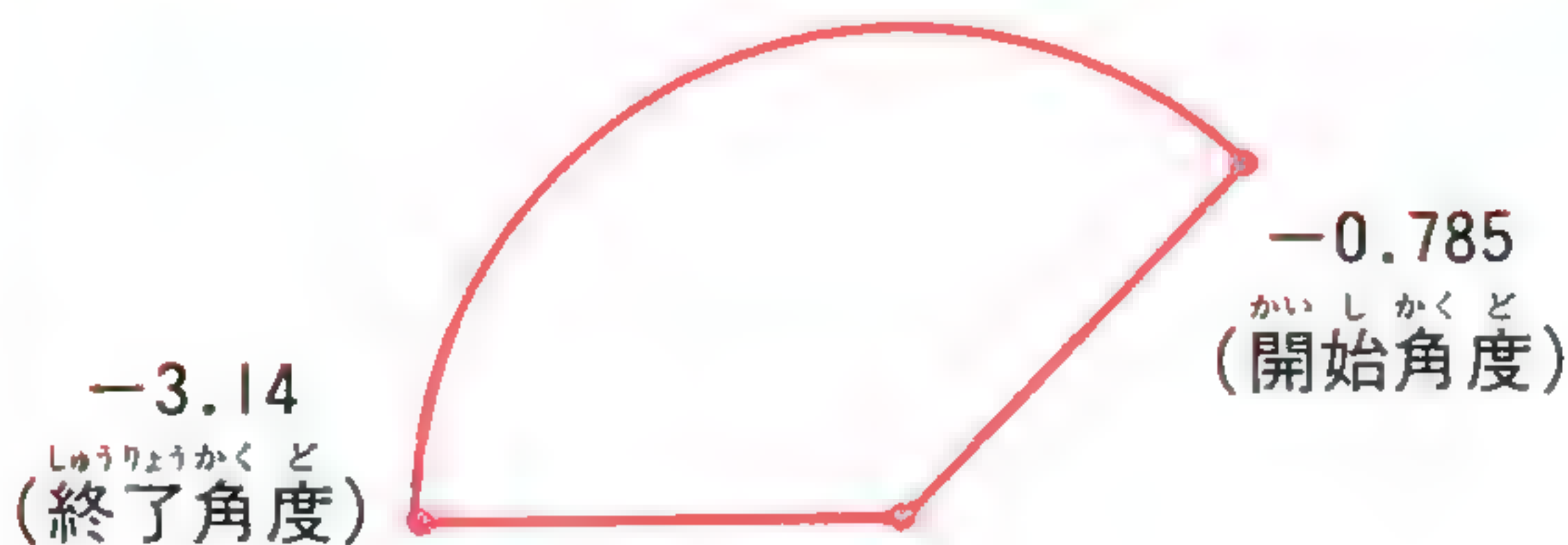
126ページの扇形を描くプログラムを少し変えて、マイナスをつけてみよう。

CIRCLE (X，Y)，半径，カラーコード，－開始角度，－終了角度

だから、次のプログラムになるね。

```
10 SCREEN 2
20 CIRCLE(130,90),50,8,-.785,
   -3.14
30 GOTO 30
```

－0.785
（開始角度）

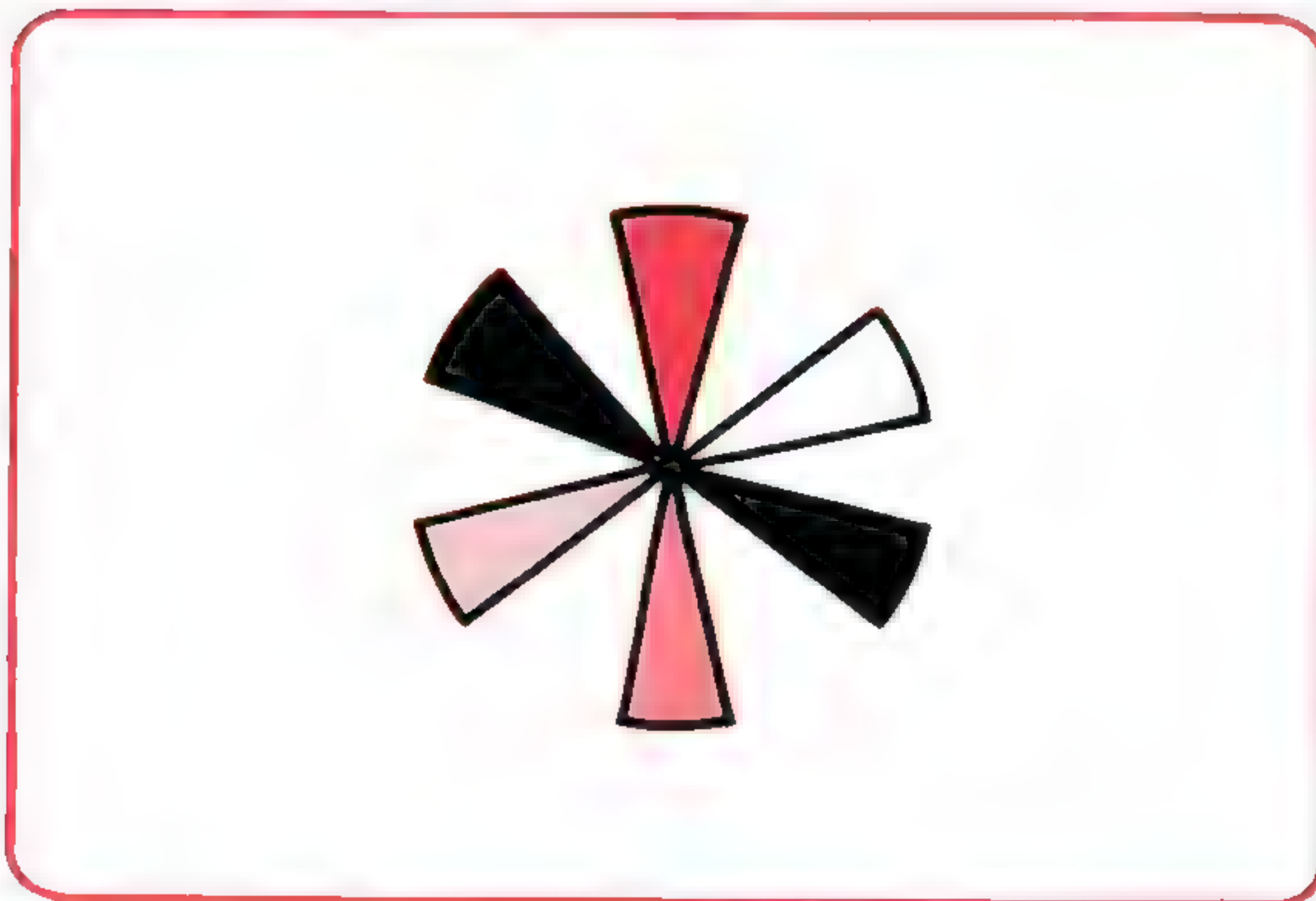
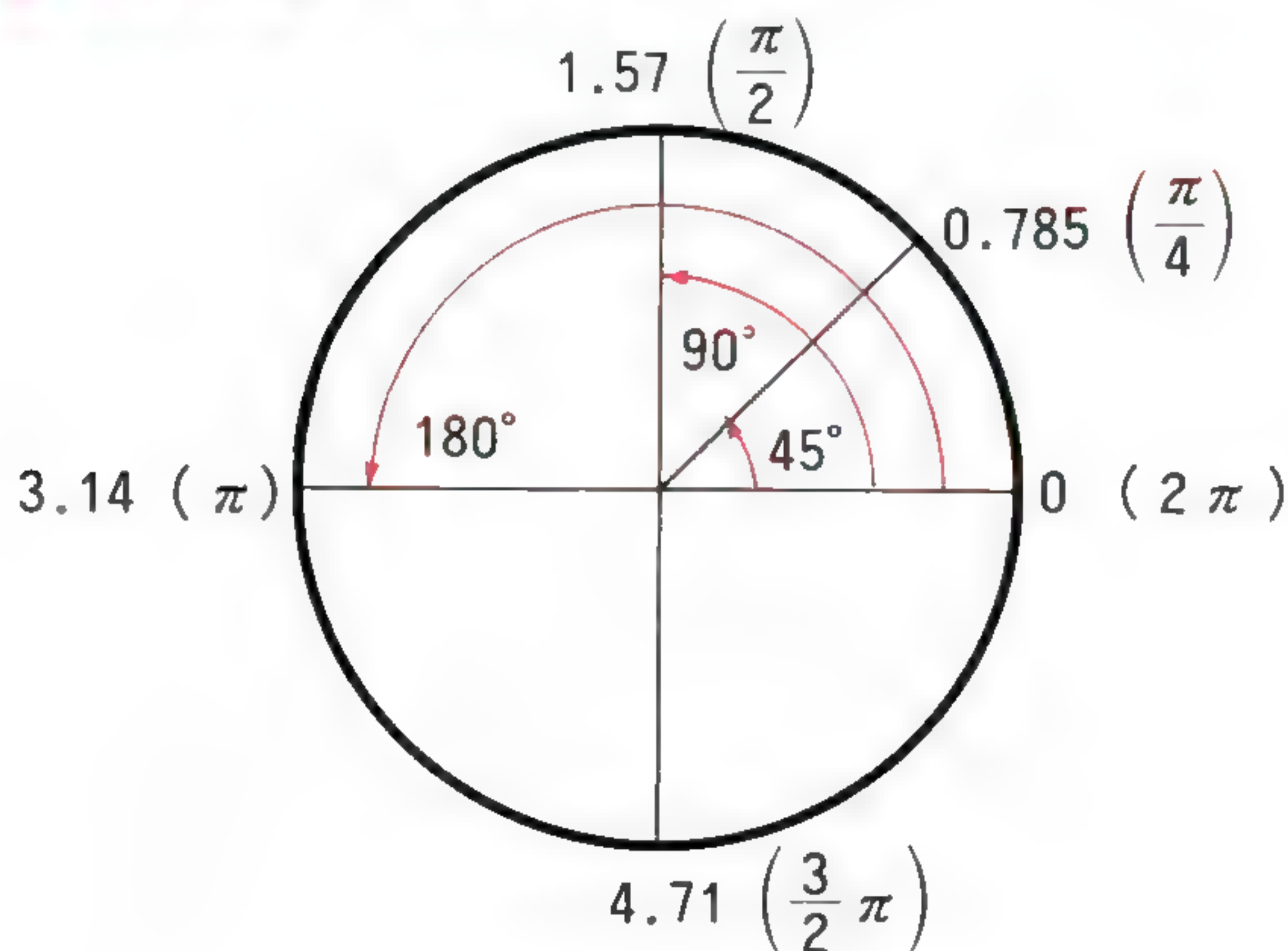
－3.14
（終了角度）

角度（ラジアン）の指定は－2π（－6.28）～2π（6.28）の範囲でないとIllegal function callのエラーになるよ

●CIRCLE文ショートプログラム　2

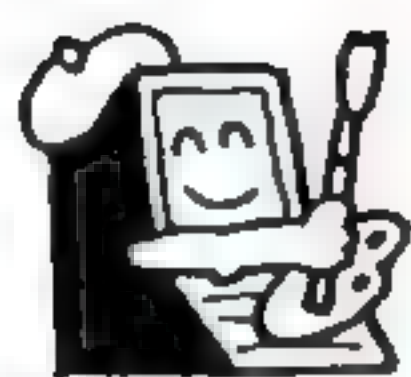
●放射線状に6個の扇形を描く

```
10 SCREEN 2
20 FOR I=0 TO5
30 R1=I*3.14/3+3.14/12
40 R2=R1+3.14/6
50 CN=I*2+1
60 CIRCLE (120,90),70,CN,-R1,-R2
70 NEXT
80 GOTO80
```

●角度を指定するラジアン値

扇形を6個それぞれ違った色で描くプログラムだ。マイナスをつけ忘れないで。

60行のCIRCLE文で、表示する色はCN、開始角度はR1、終了角度はR2と変数になっている。

扇形を描くのだから、R1、R2にはマイナス(－)をつけてある。

20行のFOR文で変数Iは0～5の6種類。だから50行の色の指定はカラーコード1、3、5、7、9、11ということだね。

縦横の比率を与えてだ円を描く

円を描いて、アーチ（円弧）を描いて、扇形を描いたら、その次はだ円形を描いてみよう。円やだ円を描くには、開始角度と終了角度はいらないよね。だから、そこは省略して、最後に「比率」というものをつけたす。

CIRCLE　（X，Y），半径，カラーコード，，，半径

実際にやってみよう。

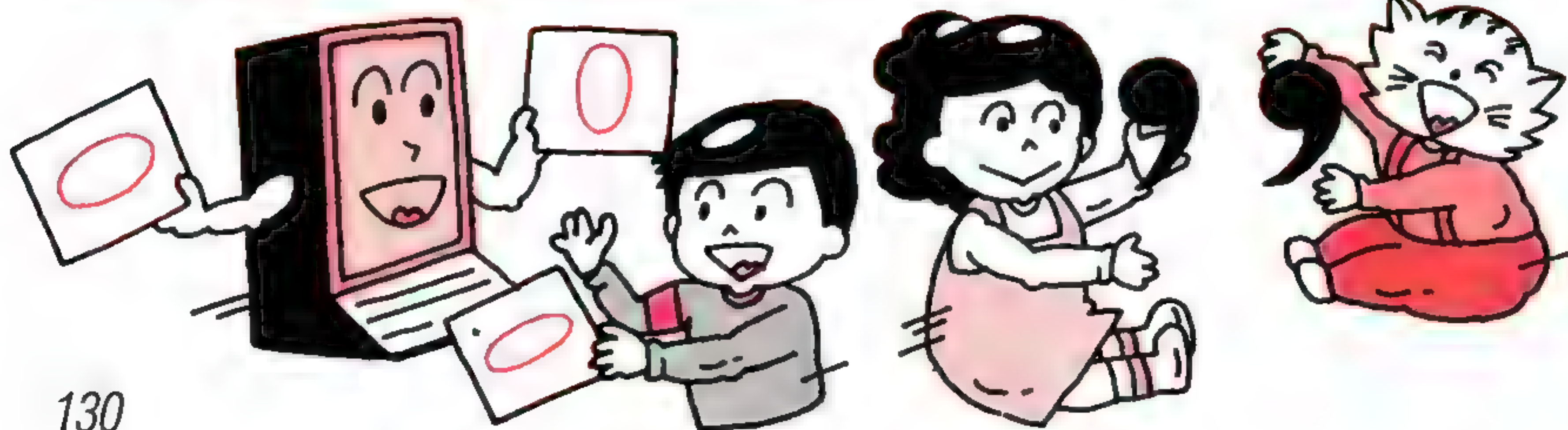
```
10 SCREEN 2
20 CIRCLE(130,90),50,8,,,2
30 GOTO 30
```

比率は（垂直方向半径）÷（水平方向半径）

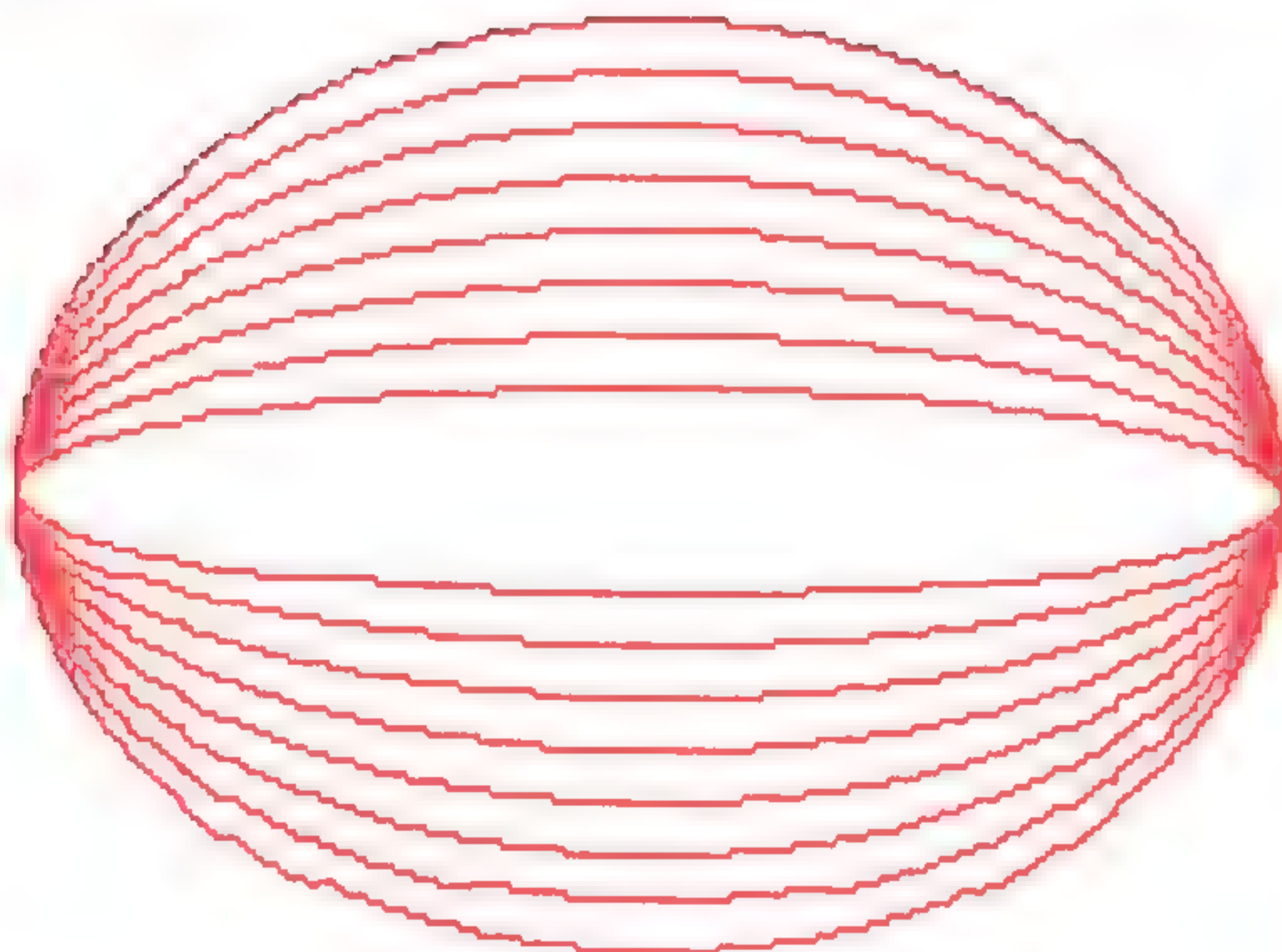
比率が1なら真円。省略してもいい

比率が1より大きければ縦長だ円

比率が1以下なら横長だ円

開始角度と終了角度を省略しても2つのコンマ（，，）は省略できない

●CIRCLE文ショートプログラム　3

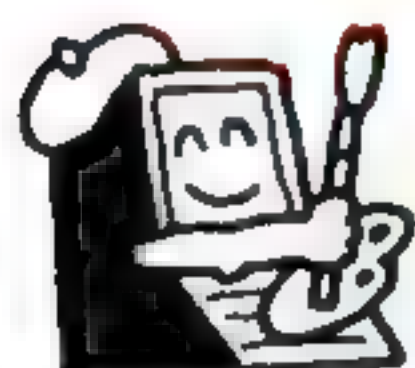
●横長のだ円形を８個描く

```
10 SCREEN 2
20 H=1
30 FOR I=1 TO15 STEP 2
40 H=H-.1
50 CIRCLE (120,90),80,I,,,H
60 NEXT
70 GOTO 70
```

50行のCIRCLE文でHの前にコンマが「，，，」と3つ並んでいるがこれは開始角度と終了角度を省略したシルシ

40行の「.1」は「0.1」のことだよ

50行のCIRCLE文で、比率を変数Hにしている。

20行でH＝1と定義し、40行でHを0.9〜0.2までひとつずつ小さくしている。H＝H−.1をH＝H＋.1にすれば、縦長のだ円になるよ。

線で囲んだ領域に色を塗る

まず出口のない図形を描いてから

PAINT命令●境界色で囲まれた場所を領域色で塗る

画面に完全に線で囲んだ図形を描いたとする。○でも△でも□でも、どんな図形でもいいのだが、たとえば、凵や∪のように出口があいている図形はいけない。完全に囲まれた部分に、べったりと色を塗るのがPAINT命令だ。

正確にいうと、指定した境界色で囲んだ場所を指定した領域色で塗るということだ。

○なら○の図形で、○の図形の線の色を境界色、○の中に塗る色を領域色というわけだ。

○の中に色インクをたらして塗っても、インクは外に流れ出さないが、凵や∪は出口があいているから、塗ったインクが外に流れ出してしまう。

LINE文や
CIRCLE文で
出口のない
図を描いて
それから
PAINT文だ

図形内にある場所を指定して

図形の中に色を塗るPAINT文は、

PAINT（X，Y），領域色，境界色

というように使う。

（X，Y）は例によって画面の場所を指定する。Xは横の位置、Yは縦の位置だ。そして、色を塗りたい図形の中なら、どこの場所を指定してもいい。

領域色は図形の外わくの色。カラーコードで指定するんだよ。

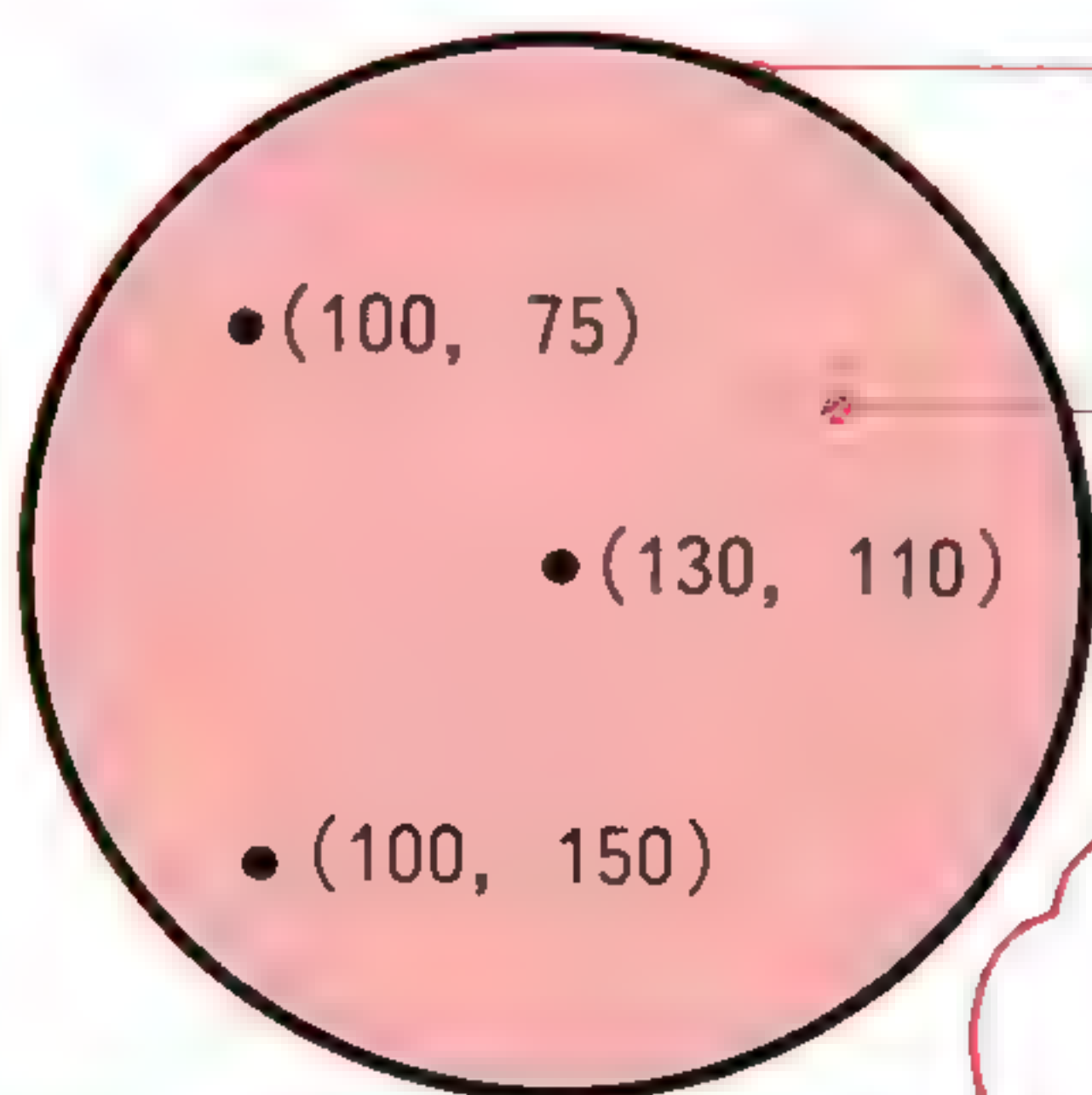
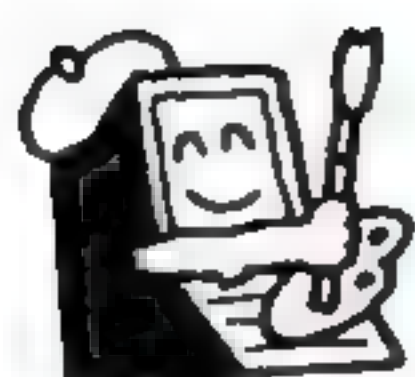
境界色が、べったり塗りたいインクの色だ。やっぱりカラーコードで指定するんだ。

SCREEN　2

境界色は使用できない。だから、境界色も領域色も指定した領域色で塗られるから、外わくはつかない。

SCREEN　3

領域色、境界色とも使えるから、外わくと内部を別の色で塗れる。

領域色と境界色

領域色を省略すると：COLOR文で指定した前景色になる。
境界色を省略すると：領域色と同じ色になる。

SCREEN2とSCREEN3

かんたんなプログラムを実行してみよう。

```
10 SCREEN 3
20 CIRCLE(130,100),50,8
30 PAINT(130,100),2,8
40 GOTO 40
```

PAINT文の(130，100)は円の内部ならどの位置の座標でもいい

カラーコード8の赤い外わく、内部が2の緑色で塗られた画面が表示されただろう。

こんどは、10行のSCREEN　3を2に変えて、CIRCLE文の最後の8を2に変えて実行してみよう。外わくはなし、ぜんぶ緑色で塗られた円が表示されたはずだ。

SCREEN　3の画面

SCREEN　2の画面

SCREEN　2　横256×縦192ドットの高解像度グラフィックモード。
SCREEN　3より目がこまかい。

SCREEN　3　横64ブロック×縦48ブロックのマルチカラーモード。
ただし、座標の指定は256×192ドットと同じ。

PAINT文ショートプログラム

三角形の中を塗りつぶす

①
```
10 SCREEN 2
20 LINE(120,60)-(40,140),2
30 LINE-(200,140),2
40 LINE-(120,60),2
50 PAINT(120,100),2
60 GOTO 60
```

20行のLINE文　40行のLINE文

(120, 100)
緑色

30行のLINE文

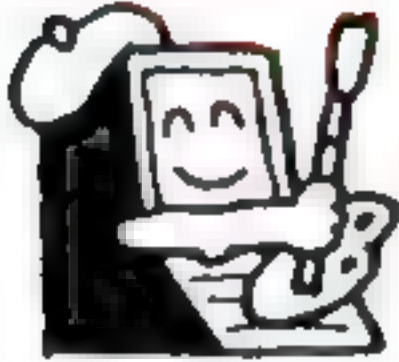
LINE文3本で三角形を描き、その内側を緑色で塗りつぶす。

SCREEN 2だから、LINE文で指定した線の色と同じ色（カラーコード2、緑色）の領域色を指定している。

日の丸を表示する

②
```
10 SCREEN 2
20 LINE(50,50)-(200,150),15,BF
30 CIRCLE(125,100),35,8
40 PAINT(125,100),8
50 GOTO 50
```

日の丸の旗を表示するプログラム。ここまでですでにおぼえてしまったグラフィック命令のLINE文、CIRCLE文、PAINT文が、みんな入っているから復習だ。

4 6 ドットに色をつけて表示させる

どの位置のドットを表示させるか

PSET命令●指定した座標の点を表示

グラフィック画面には、SCREEN　2と、SCREEN　3の2通りがあるのだったね。そして、SCREEN　2は横256、縦192のドット（点）がうめこまれているし、SCREEN　3は横64、縦48のブロックに分かれているということだった。

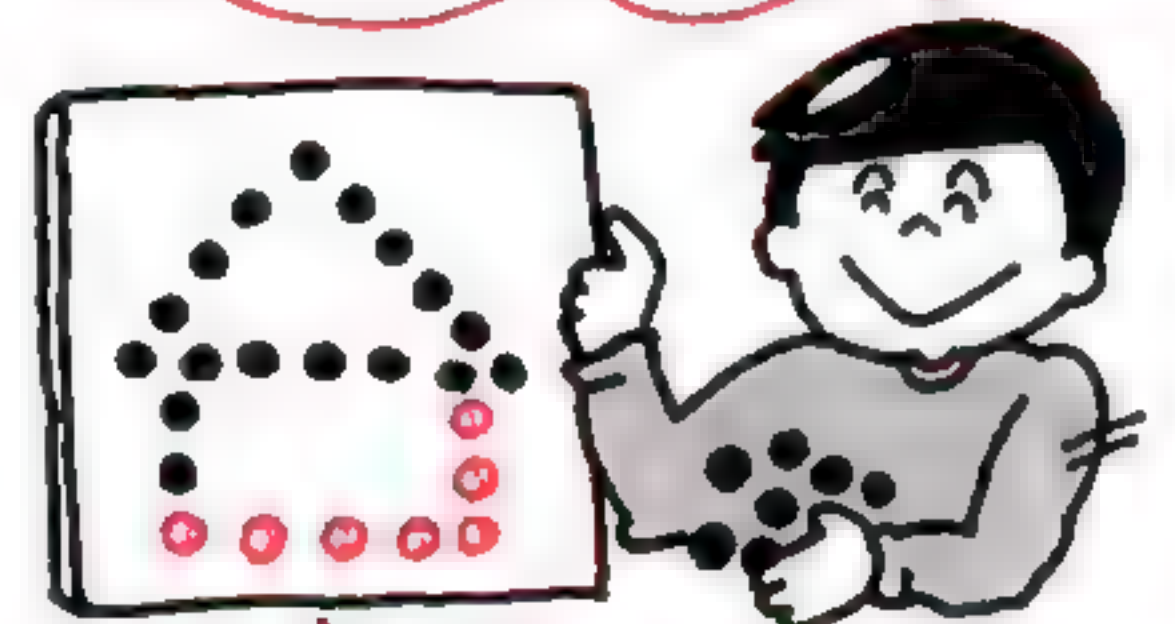
このドットやブロックを指定して、あそこのドットを表示させなさい、ここのブロックを表示させなさいというのが、PSET命令なのだ。それも、お好みの色で表示させることができるのだ。

PSET　(X，Y)，カラーコード

(X，Y)の位置のドットまたはブロックを、カラーコードの色で表示させるというわけ。

```
10 SCREEN 2
20 PSET (120,90),8
30 GOTO 30
```

画面の中央のあたりに、小さな小さな赤い点が表示されたはずだよ。点が小さいので、赤い点には見えないかもしれないが。

ドットやブロックは絵を表示させるもとだから画素というんだよ

座標（120，90）の位置にカラーコード8の赤い点を表示した

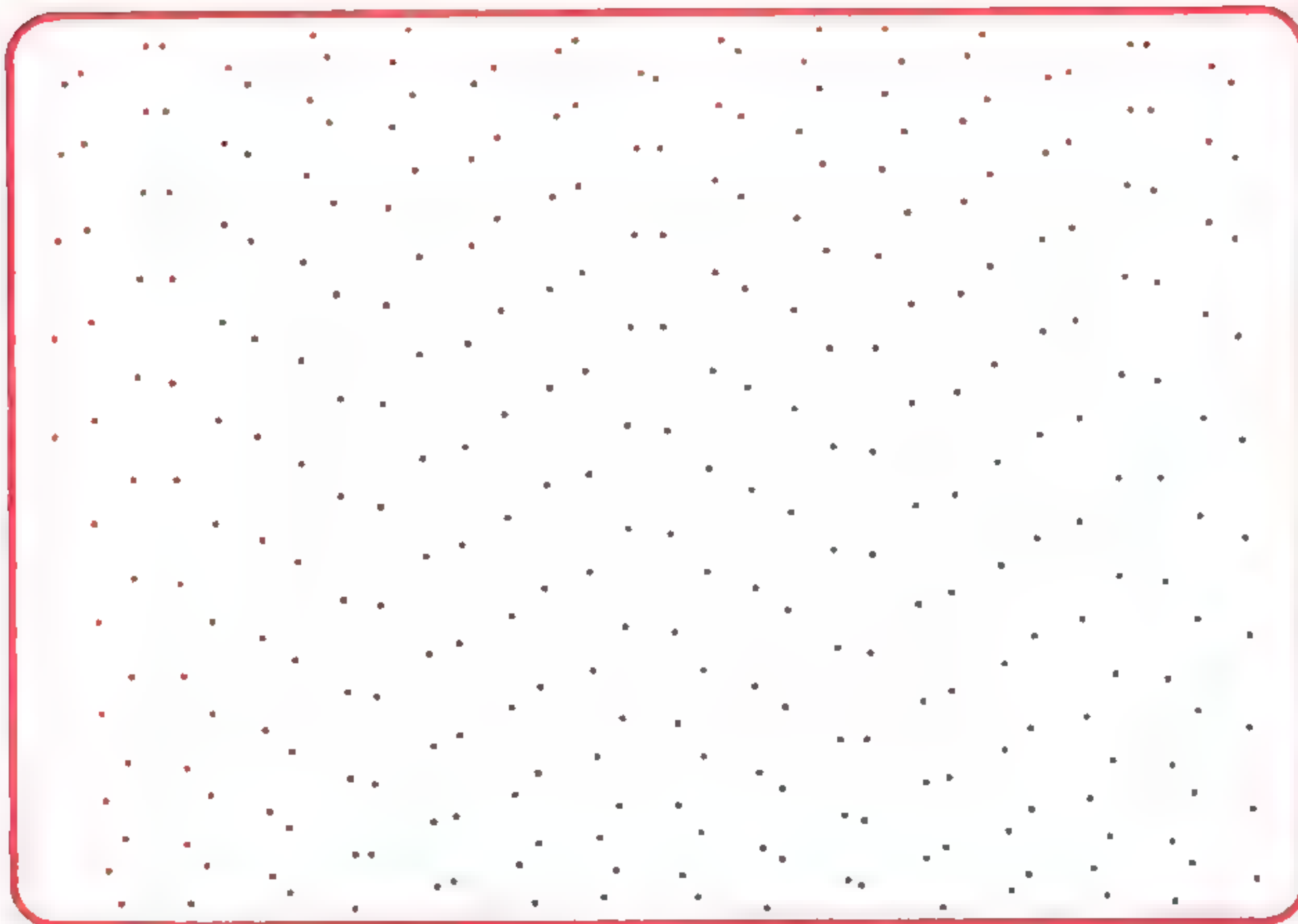
●
赤い点

●PSETショートプログラム

●よく見ればSINカーブの点点点……

```
10 COLOR 15,1
20 SCREEN 2
30 CN=2
40 FOR X=1 TO255 STEP .4
50 Y=90*SIN(X/2)
60 IF 8/X=INT(8/X) THEN CN=CN+1
70 IF CN=16 THEN CN=2
80 PSET(X,Y+100),CN
90 NEXT
100 GOTO100
```

STEP .4は
STEP 0.4の
ことだよ

PSETは
かくれたドットを
引っぱり出して
目で見えるように
するんだな

40行は横方向（X軸）の値を1から255まで変化させている。STEPのあとの数字を小さくするほど、模様はこまかくなるよ。

50行は、SIN関数を使って縦方向（Y軸）の値を計算している。

60行、70行は色を決めている。ちょっと複雑な計算だから、いまはわからなくても気にしない。

点を消す方法で点を表示させる方法

使い方しだいでPSETと同じ

PRESET命令●指定した位置の点を消す

点を表示させるのがPSETなら、表示されている点を消すのがPRESETだ。

PRESET　(X，Y)，カラーコード

(X，Y)で指定した位置にある点（ドット）をカラーコードの色で消すという働きをする。

ところが実際には、消したはずの点が、表示されたようにみえる使い方があるからおもしろい。

```
10 SCREEN 3
20 LINE(20,10)-(100,50),8,BF
30 PRESET(60,30)
40 GOTO40
```

画面に赤色の箱が表示されたあと、その中央に暗い水色が表示されたと思うよ。なぜ？

起動時の背景色は水色。その上に赤で塗った箱がある。

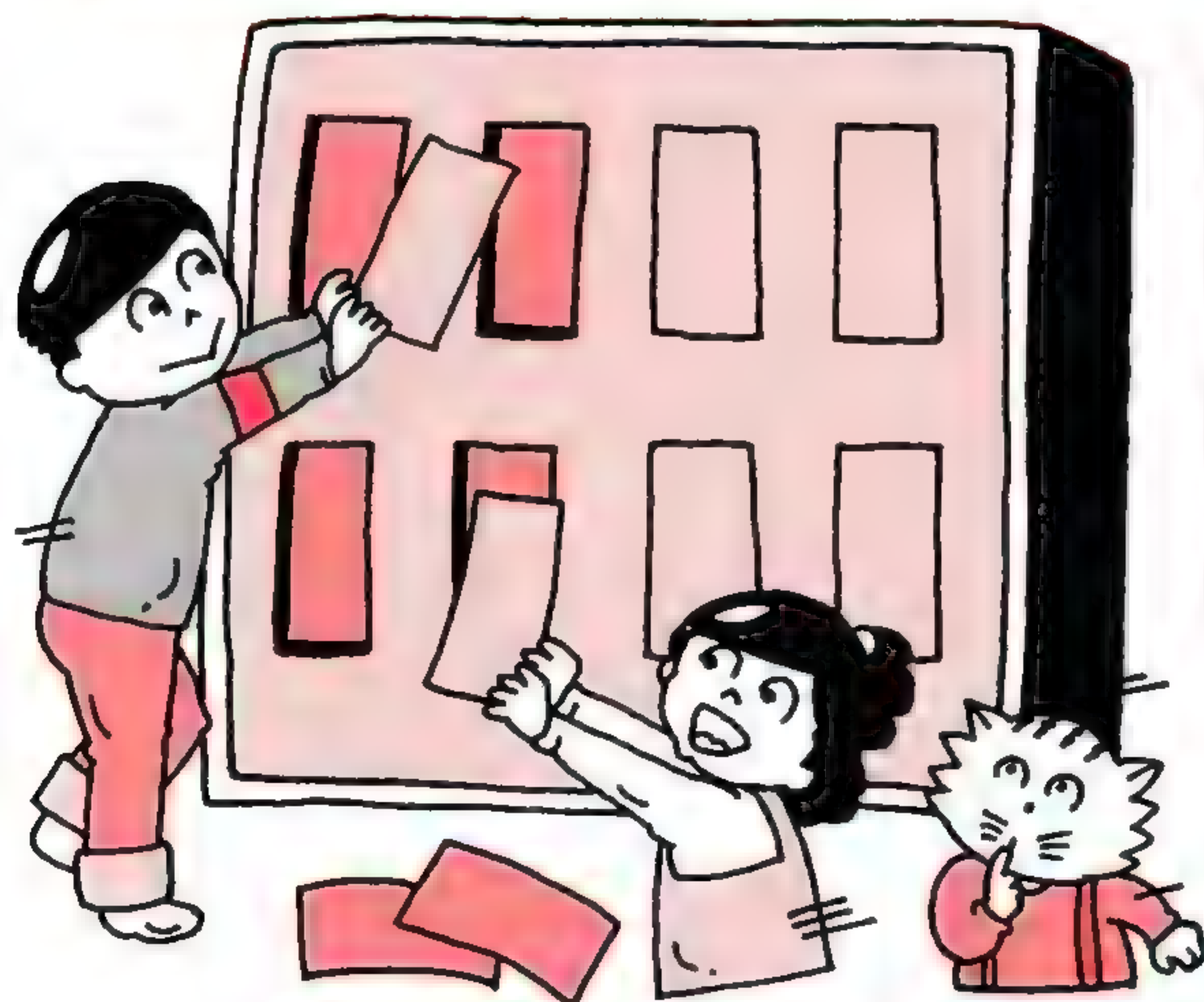
その赤をPRESETで消すから、下地の水色が表面に出てきた。とこう考えれば、わかるだろう。

●PRESETショートプログラム

●どんどん点を消していく

```
10 SCREEN 2
20 LINE(50,50)-(200,150),2,BF
30 FOR I=50 TO150
40 FOR J=50 TO200
50 PRESET(J,I)
60 NEXT J,I
70 END
```

20行のLINE文で描いた緑色の長方形。50行にあるPRESET文は、四角の緑色を上から順番にはぎとっていく。

緑色をはぎとられた部分は、下に塗ってある色（水色）が表面に出て新しい点を表示していくようだ。

パソコンのペン描き

方向も長さも思いのまま

DRAW命令●画面に自由な図形を描く

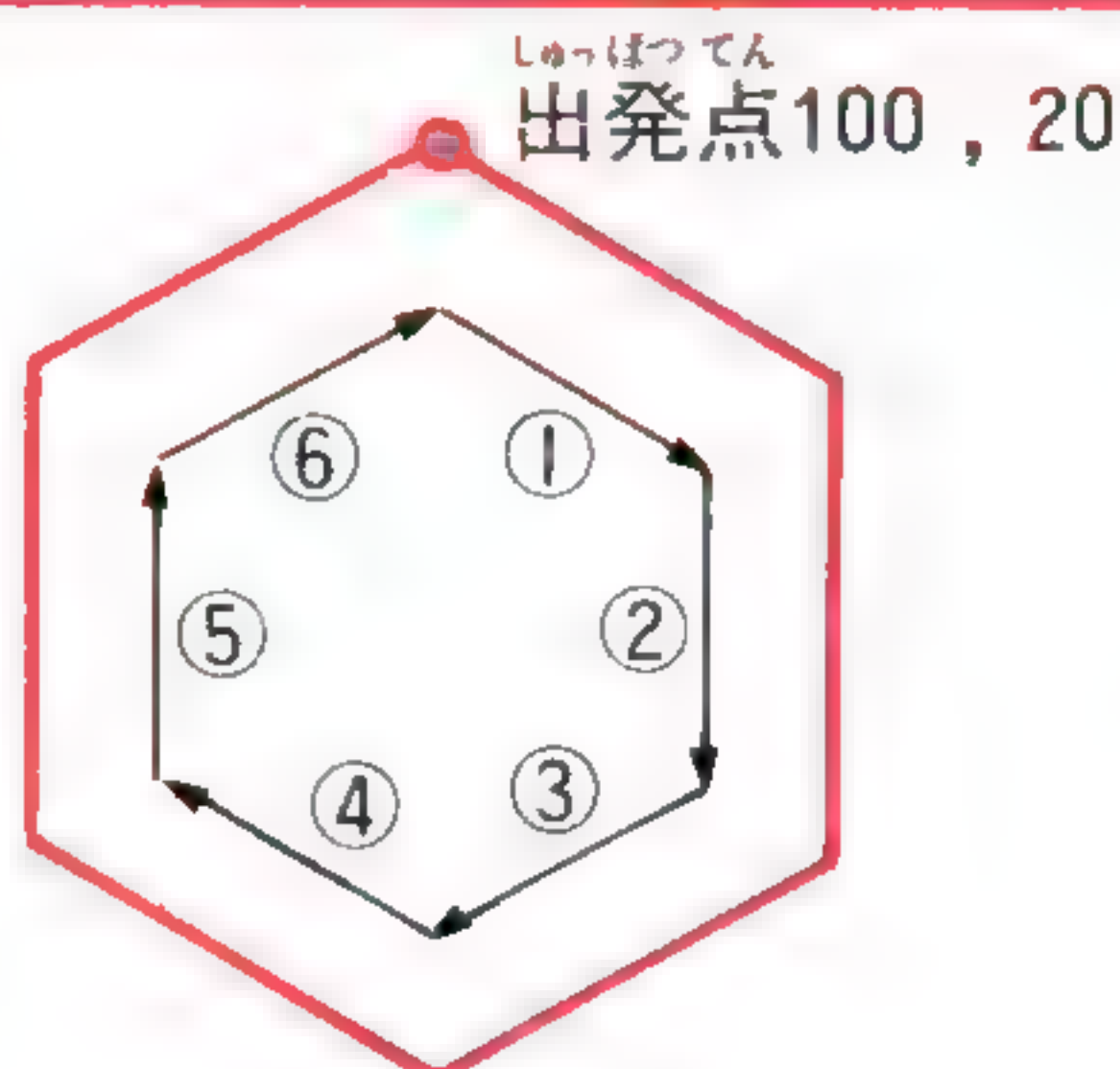
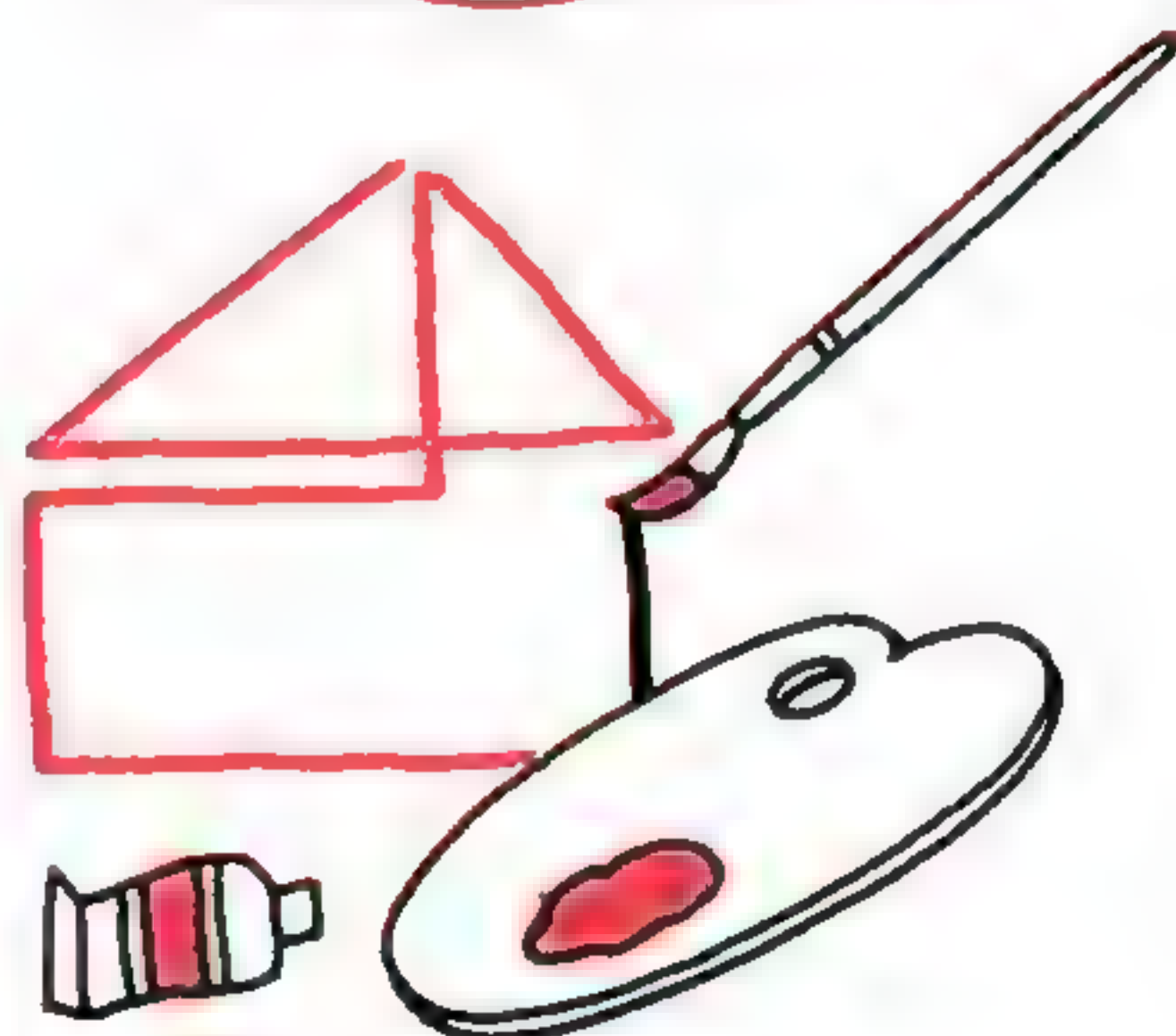
ディスプレイの中に絵の具とペンがかくれていて、それを使って自由に絵を描くのが、DRAW命令だ。ペンを思ったとおりに動かせるから、どんな絵でも描けるよ。

DRAW　"文字式"

という文型だから、「文字式」のところで、ペンを動かす方向や長さ、色などを指定してやる。

次のプログラムを入力して実行してみよう。

DRAW命令は
ひと筆描きの要領だ
ひと筆描きでなくても
なんでも描けるけどね

```
10 SCREEN 2
20 DRAW"C8BM100,20"
30 DRAW"F50D50G50H50U50E50"
40 GOTO40
```

20行のＣ８はカラーコード８の赤。Ｂは線を描かずにペンを次の位置まで移動。M100, 20は、ペン画を出発させる位置だ。

30行のDRAW文でローマ字は進む方向、数字は進むドット数。

●DRAW文の文字式

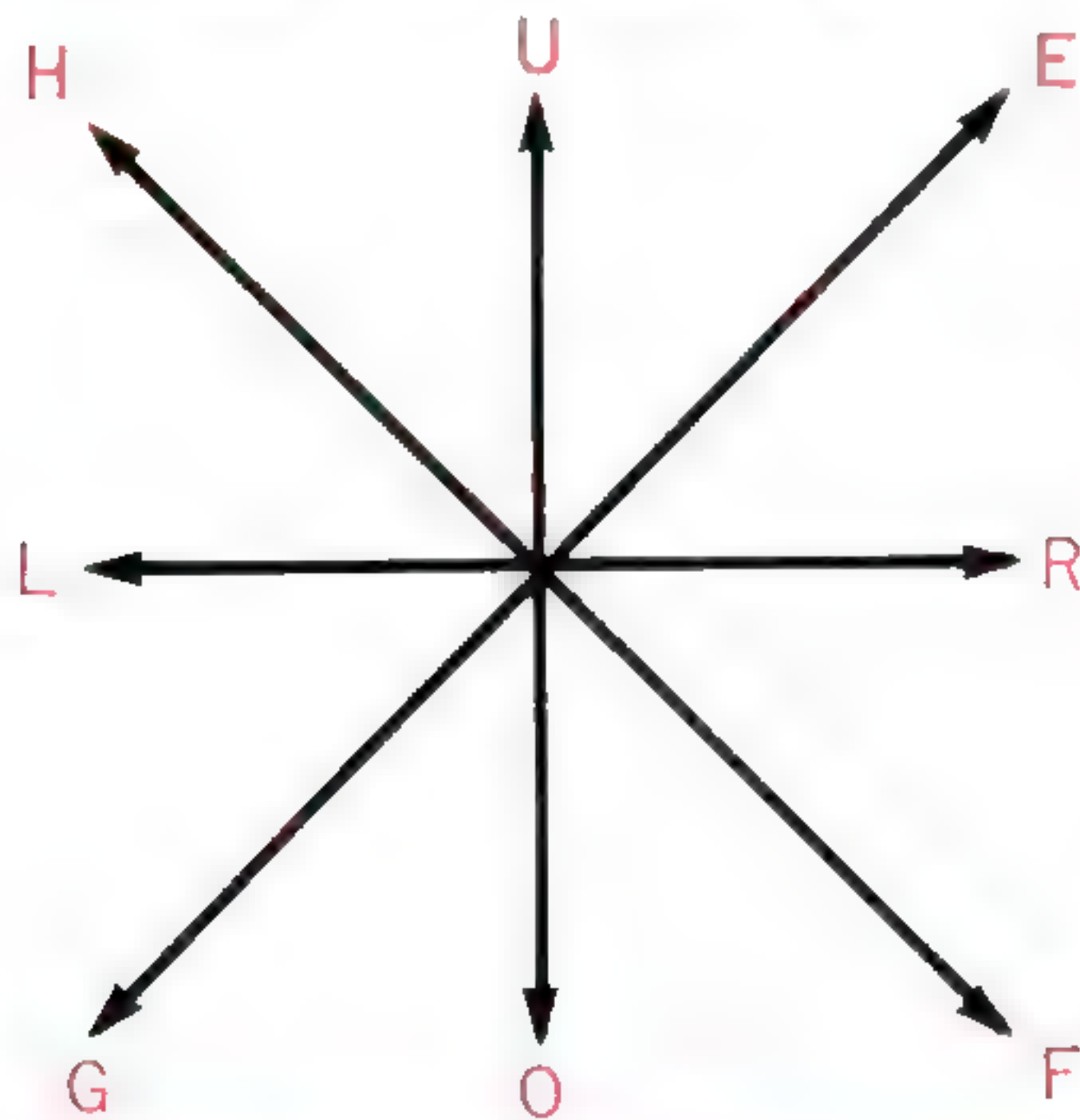
前のページのプログラムを説明してみよう。

20　DRAW "C 8 BM100, 20"

- ペンの出発点の座標。横100、縦20の位置だ
- ペンを移動させるが、線は引かない
- 描く線の色。カラーコードで指定する

30　DRAW "F50D50G50H50U50E50"

- ローマ字はペンの進む方向。数字は進む長さ

●ペンの移動方向と移動量

ペンは８方向に進められる
U、E、R、F、O、G、L、Hの８方向だよ

方向の次には引く線の長さを指定だ
U50なら上の方向に50ドット
F50なら右下の方向に50ドット進む

同じドットを指定しても
水平、垂直の線より
斜めの線は実際の距離は長いよ

●たくさんのMSXを描く

```
10 SCREEN 2
20 DRAW"BM30,50C8S2A0"
30 GOSUB150
40 DRAW"BM70,50C2S4"
50 GOSUB150
60 DRAW"BM130,50C15S8"
70 GOSUB150
80 DRAW"BM50,130C7S4A1"
90 GOSUB150
100 DRAW"BM150,100C9A2"
110 GOSUB150
120 DRAW"BM180,100C13A3"
130 GOSUB150
140 GOTO 140
150 DRAW"M+4,-16R4M+2,10M+2,-10R4M+3,12"
160 DRAW"R9U2L7U10R18M+3,5M+3,-5R5
170 DRAW"M-5,8M+5,8L5M-3,-5M-3,5L5M+5,-8M-4,-4"
180 DRAW"L10D3R6D9L15M-2,-10M-2,10"
190 DRAW"L4M-2,-10M-2,10L5"
200 FOR I=1 TO1000:NEXT
210 RETURN
```

DRAW命令で〝MSX〟という文字を、いろいろ変化させて表示している。

表示場所や文字の大きさ、あるいは表示の向きを変えたり、いろいろできるということをわかってくれればいいと思うよ。

〝MSX〟という文字を書かせるDRAW命令は、150行～190行までのDRAW文。

ただ、このプログラムは、〝MSX〟という文字を表示させるというだけじゃない。どんなふうに表示させるかということを、20行～120行のとびとびのDRAW文で説明している。

座標の指定にマイナス(－)やプラス(＋)がついているのは、前の座標からマイナスやプラス方向にずれた点という意味だ。

コマンドのSはスケールファクタマクロ命令
Aは回転マクロ命令だ

グラフィック画面に文字を描く

開いて書き、書いたらCLOSE

OPEN命令●ファイルを開く

グラフィック画面に文字を書くときには、どうしてもしなければならないことがある。次のプログラムを実行してみよう。画面のほぼ中央に赤で"MSX"、緑色で"BASIC"と表示されただろう。

PRINT＃は
ファイルにデータを
書き出す命令だ

```
10 SCREEN 2
20 OPEN"GRP:"AS #1
30 COLOR 8
40 LINE(70,50)-(200,120),15,BF
50 PRESET(100,80)
60 PRINT #1,"MSX ";
70 COLOR 2
80 PRINT #1,"BASIC"
90 CLOSE #1
100 COLOR 15
110 GOTO 110
```

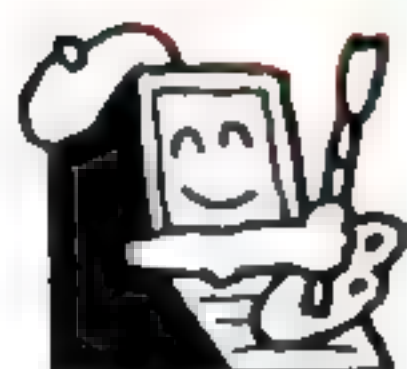
グラフィック画面に
白い箱を描いて、
その中に文字を表示
させるプログラムだ

20行のOPEN文をよく見てほしい。

OPEN "GRP:" AS #1

これは、グラフィック画面に文字を表示するときは、かならず実行する約束みたいなものだから、そっくりそのまま暗記してしまおうよ。

閉いたものはCLOSEで閉じることになる。

4 10 パソコン・ミュージック

PLAY命令と約束ごとをおぼえて

PLAY命令● "　"で囲んだメロディーを演奏する

パソコンに音楽を演奏させる命令が、PLAY だ。PLAYのあとに、"　"(ダブルクォーテーション) で囲ってメロディーや、演奏に必要な約束ごとを書くんだ。

ピアノや笛やバイオリン、ぼくたちが楽器を弾くときは楽譜のオタマジャクシを見て演奏する。

パソコンは楽譜は読めないから、かわりにコンピュータ用のミュージック言語という特別なことばを使って、音楽を演奏させるんだ。

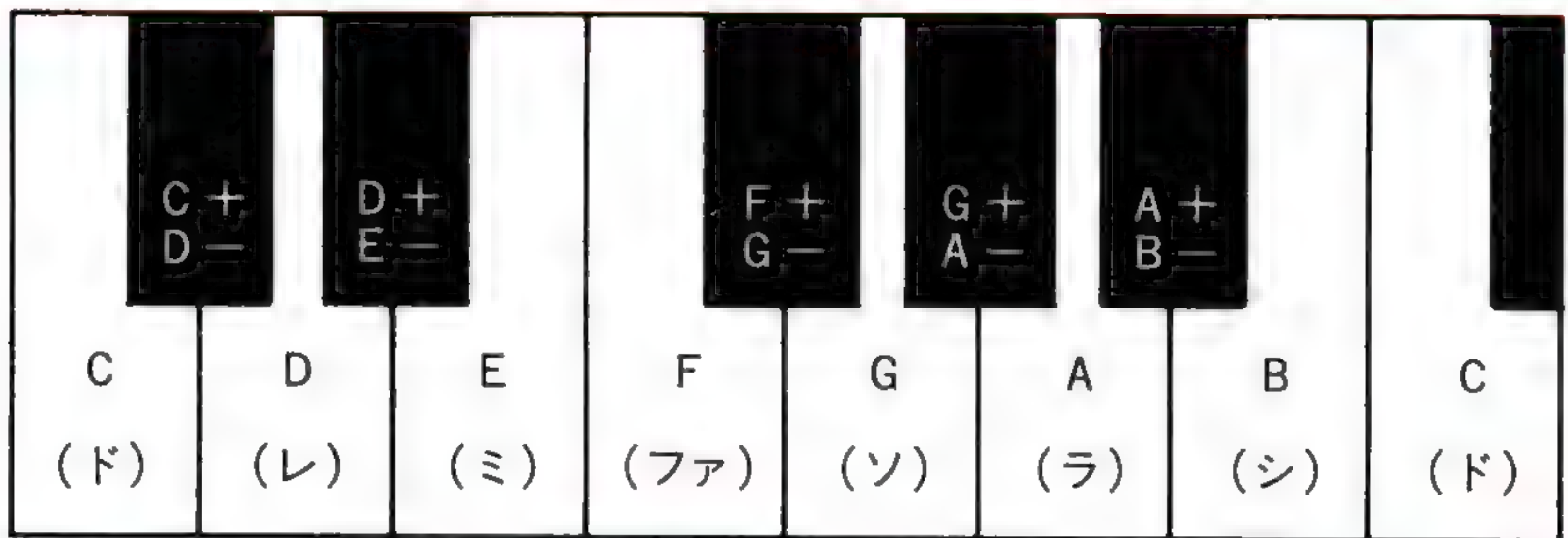
ちゃんとした音楽が演奏されるためには、音階、音の長さ、オクターブ、テンポ、休み、音の大きさなど、いろいろな要素がある。これらをすべてパソコンがわかるように、ミュージック言語になおしてやらなければならない。

たとえば、音階の「ドレミファソラシ」は、ミュージック言語では「CDEFGAB」、4分休符はR4というぐあいだ。

まず、この約束ごとをおぼえてしまおう。

MSXは、最高3重和音まで演奏できる PLAYのあとに "　"を3つ，(コンマ) でつなげばいいんだ

きみが音ちでも心配ないよ。正しい命令を与えてやれば、パソコンがちゃんと演奏してくれるさ

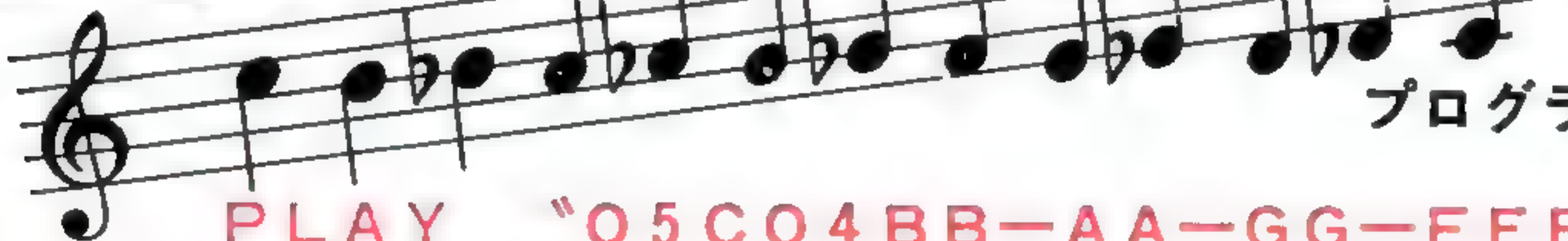
●CDEFGAB

ドレミファソラシは、CDEFGAB

ド　レ　ミ　ファ　ソ　ラ　シ
(C) (D) (E) (F) (G) (A) (B)

プログラムにすると、

PLAY　"CDEFGAB" だ

プログラムにすると、

PLAY　"CC+DD+EFF+GG+AA+B"

プログラムにすると、

PLAY　"O5CO4BB−AA−GG−FEE−DD−C"

ドレミファソラシの音階(おんかい)はCDEFGAB。半音上(はんおんあ)げるときは音階(おんかい)のうしろに+(プラス)または#、下(さ)げるときは−(マイナス)記号(きごう)をつける。

●音の長さと休みの長さ

音の長さ

全音符	𝅝	L 1
2分音符	𝅗𝅥	L 2
4分音符	♩	L 4
8分音符	♪	L 8
16分音符	𝅘𝅥𝅯	L 16
32分音符	𝅘𝅥𝅰	L 32
64分音符	𝅘𝅥𝅱	L 64

休みの長さ

全休符	𝄻	R 1
2分休符	𝄼	R 2
4分休符	𝄽	R 4
8分休符	𝄾	R 8
16分休符	𝄿	R 16
32分休符	𝅀	R 32
64分休符	𝅁	R 64

音の長さは、アルファベットのLと数字であらわす。Lは英語のLength（長さ）の頭文字だ。

音の長さはL1からL64まである。数が大きくなるほど、短い音になるよ。

書き方は2つあって、どちらもOK。

（例）4分音符のソ

L4C　または

C4

符点音符には、.（ピリオド）をつける。

（例）付点4分音符のソ

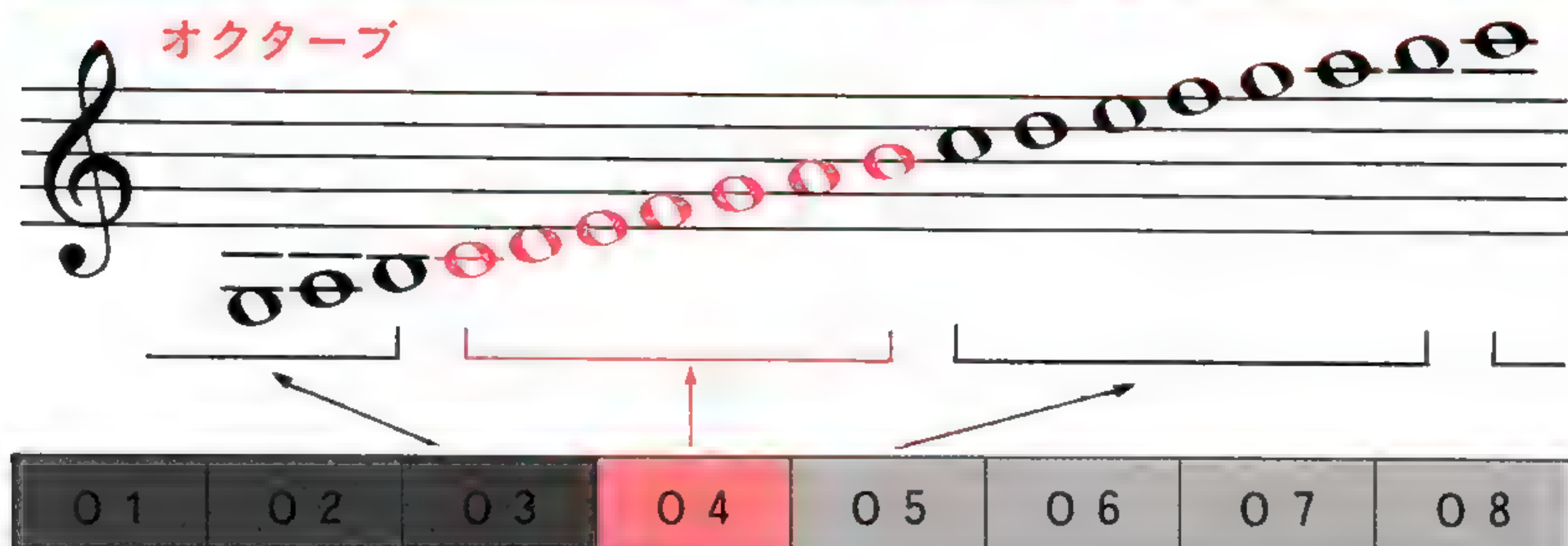
♩. ＝ ♩ ＋ ♪……………G4.

付点2分音符のソ

𝅗𝅥. ＝ 𝅗𝅥 ＋ ♩……………G2.

Lの記号をRに変えると、休みの長さだ。RはRest（休み）の頭文字で、Lと同じく、数が大きいほど休みの長さは短くなるよ。

パソコンに何も指定しなければ、音の長さはL4、休みの長さはR4にセットされている。

●オクターブ、テンポ、ボリューム

低いオクターブ←――中心のオクターブ――→高いオクターブ

音階はドレミファソラシの７つの音からできているが、同じドでもすごく高いドもあれば、低いドもあるね。

この音の高さを指定するのがオクターブ（Octave）の頭文字、Oだ。

いちばん低いオクターブO１から、いちばん高いオクターブO８までが出せる。

テンポ

曲の速さを指定するのが、テンポ（Tempo）の頭文字、Tだ。

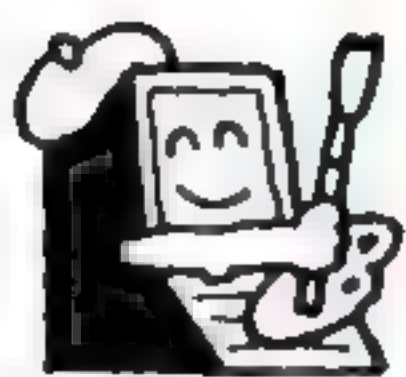
Tは、32から255までの数字で指定し、数字が小さいほどゆっくりしたテンポ、大きくなるほど速いテンポで演奏するんだ。

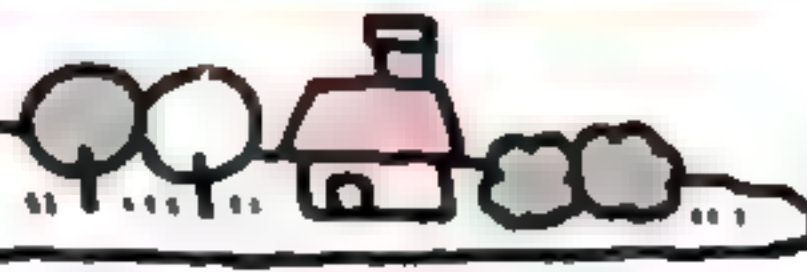
ボリューム

ボリューム（Volume）は音の大きさだ。Vで指定して、そのうしろに０から15までの数字をつける。数字が大きくなるほど音は大きくなる。

```
10 PLAY "O3 CDEFGAB"
10 PLAY "O4 CDEFGAB"
10 PLAY "O5 CDEFGAB"
```

オクターブを３つ変えてみた。同じようにVやTに変えて、違いをためしてみよう。

> パソコンに何も指定しなければ、それぞれ、O４、T120、V８にセットされている。

ミュージック・プログラミング

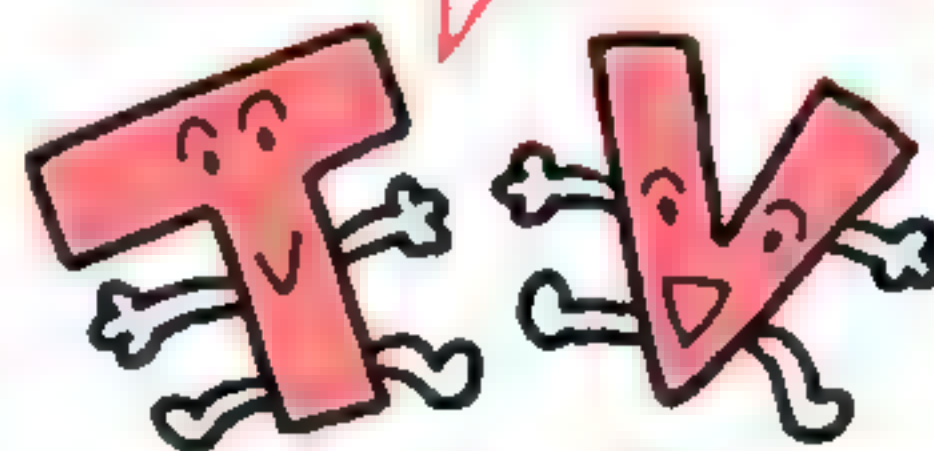
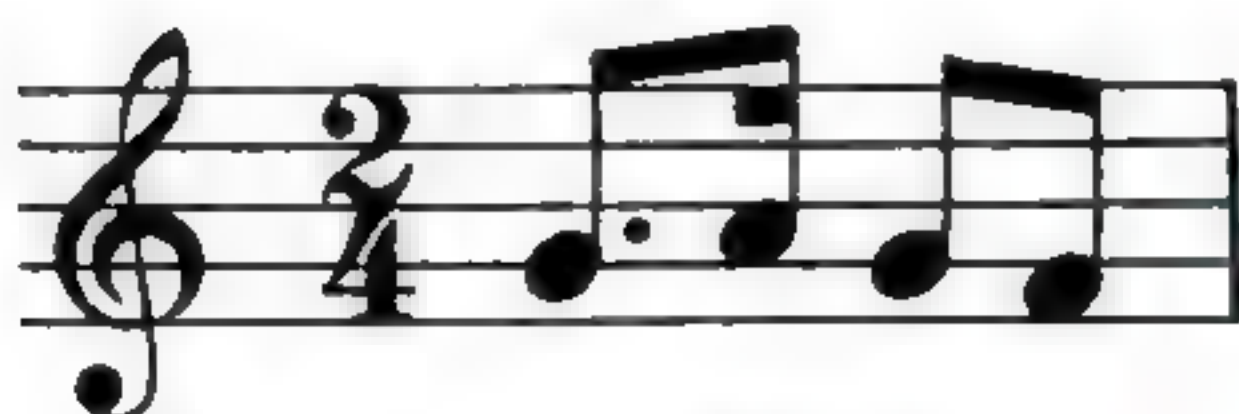
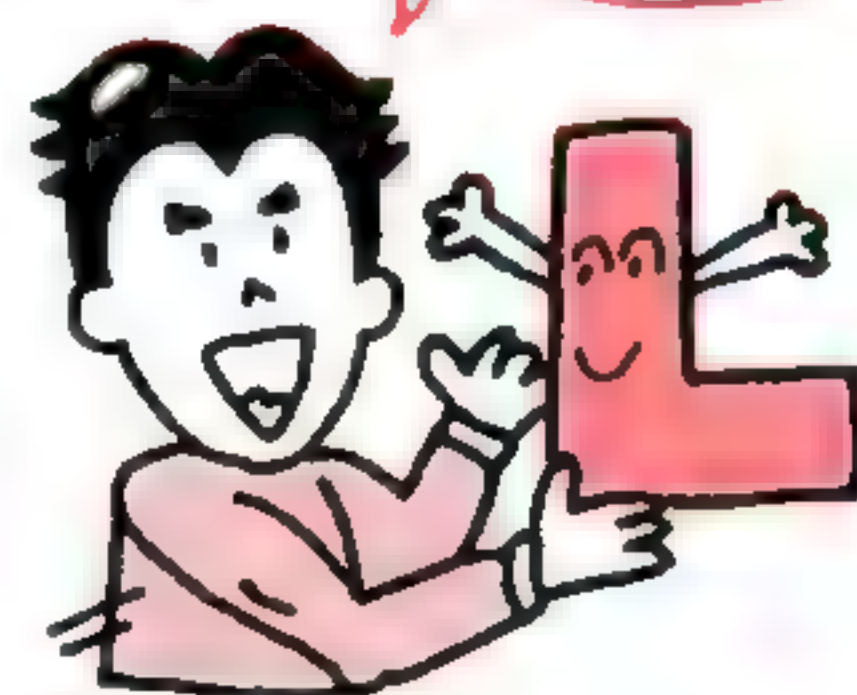
♩=180

ミュージック言語をまとめて説明したので、実際にかんたんな曲を演奏してみよう。イギリス民謡の「ロンドン橋」だ。

譜を見ると、8分音符と16分音符、4分音符、4分休符の組み合わせだ。

まず、テンポ、オクターブ、ボリュームの確認だ。曲の頭にこれを設定しておこう。

```
10　CLS
20　PLAY　"T180　O4　V8"
```

と一行にまとめて設定しておこう。

次は、30行に1小節目をプログラミングしよう。

```
30　PLAY　"L8G.L16AL8GF"
```

```
40　PLAY　"EFL4G"
```

●音の長さ、オクターブは変わるたびに指定する

残りを自分でプログラミングしてみよう。

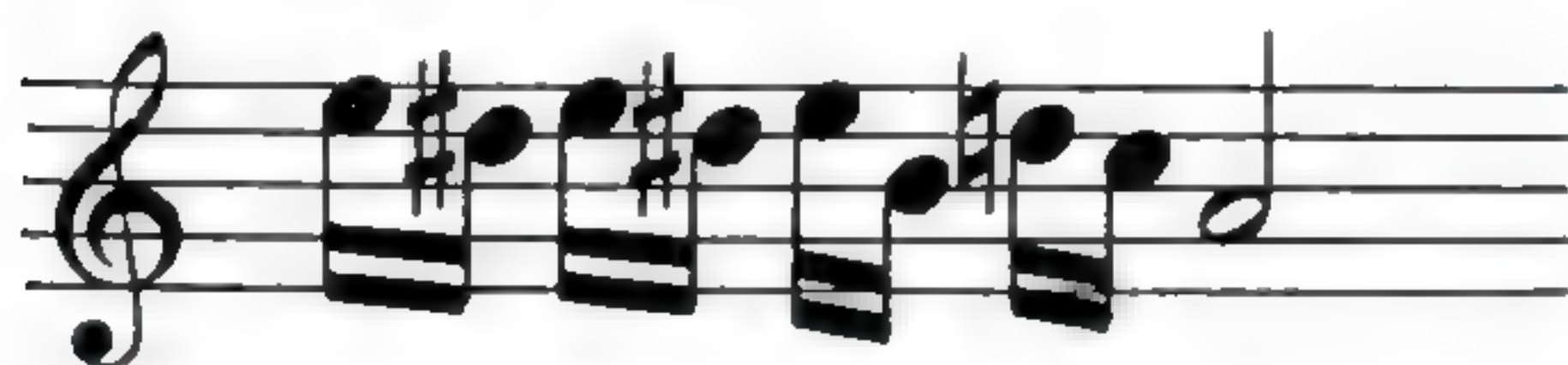
```
10 CLS:LOCATE9,10
20 PRINT"ﾛﾝﾄﾞﾝ ﾊﾞｼ"
30 PLAY"T180O4V8"
40 PLAY"L8G.L16AL8"
50 PLAY"GFEFL4G"
60 PLAY"L8DEL4F"
70 PLAY"L8EFL4G"
80 PLAY"L8G.L16AL8GF"
90 PLAY"EFL4G"
100 PLAY"DGL8ECR4"
110 GOTO 30
```

プログラミングの注意1

音の長さは、変わるたびに指定しなおす。指定を忘れると、前に指定された長さで演奏される。

プログラミングの注意2

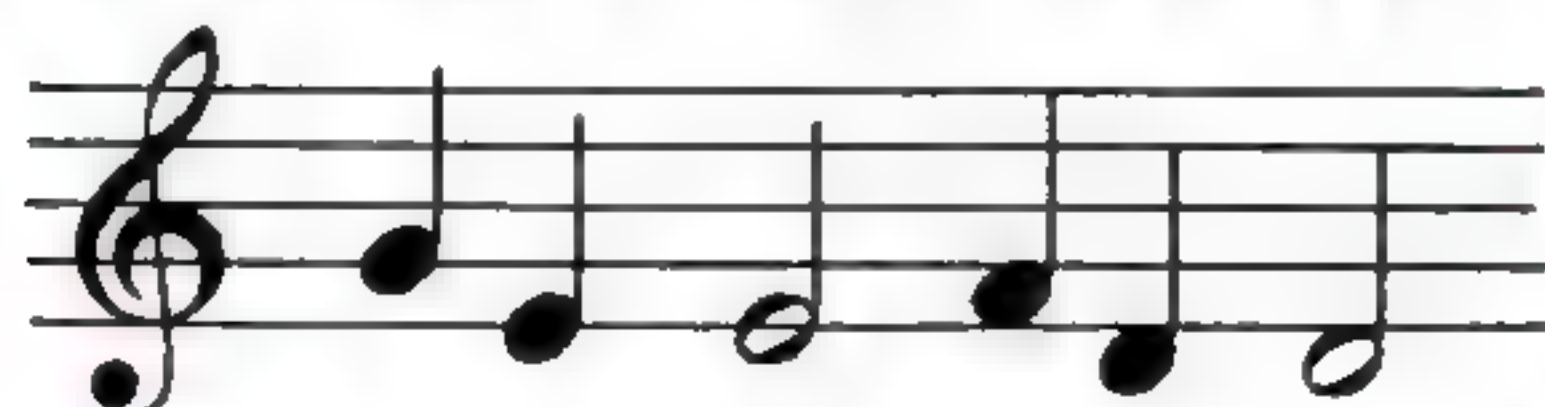
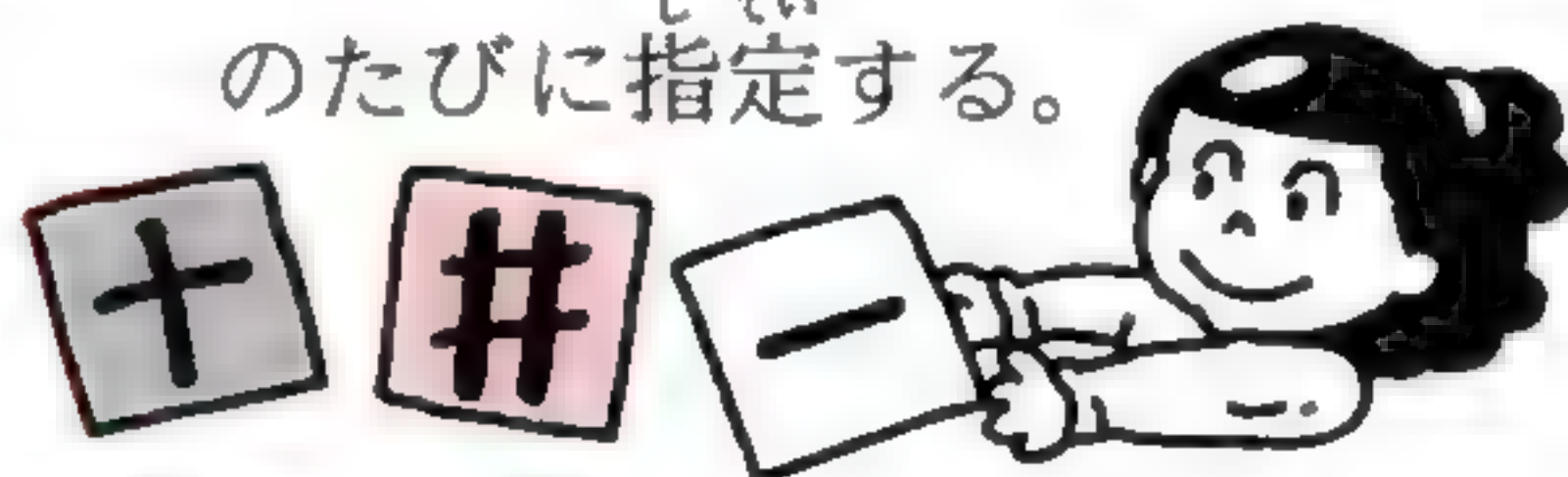
オクターブも、変わるたびに指定しなおす。指定を忘れると、前に指定されたオクターブで演奏される。

PLAY　"L4GAGO5CO4GAL2GL4AAGA"

PLAY　"L16O5ED+ED+EO4BO5DCO4L2A"

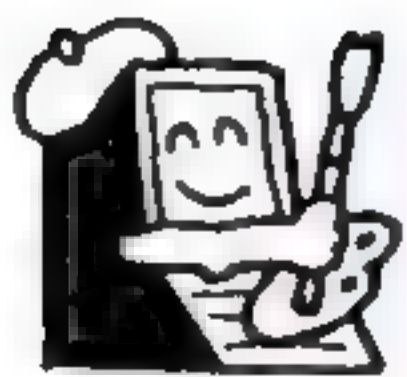
プログラミングの注意3

半音上げる（＋または＃）、半音下げる（－）指定は、そのたびに指定する。

休符を利用して同じ音を区切る

パソコンは楽器とちがって、同じ音を切って演奏することができない。同じ音が続くときは、休符を使って区切ってやろう。

PLAY　"GEE2L4FDD2"

⬇

PLAY　"GE8R8E2L4FD8R8D2"

3つの音まで同時に出せる

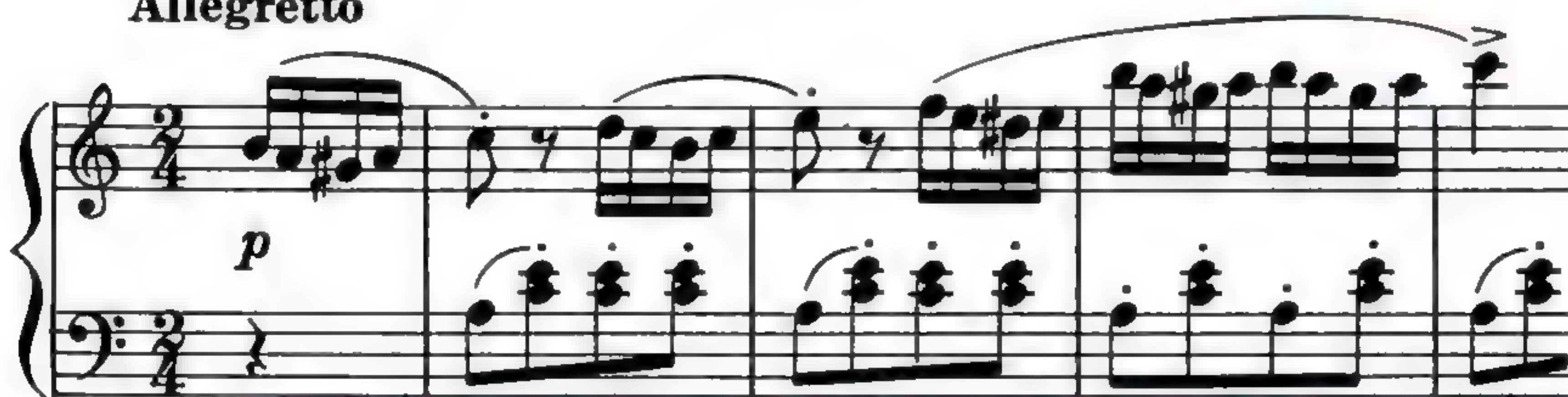
ひとつの音なら、PLAYのあとに〃　〃がひとつ、2つの音を同時に出すなら〃　〃，〃　〃、3重和音なら、〃　〃，〃　〃，〃　〃だ。〃　〃と〃　〃は、かならず，（コンマ）でつなぐこと。4つつなぐとエラーだよ。

2重和音なら

PLAY　"CDE"，"EFG"

3重和音なら

PLAY　"CCO3B"，"EFG"，"GFG"

次の楽譜は、モーツァルトのトルコ行進曲の出だしだ。復習のつもりでプログラムにしてみよう。

プログラム100行　110行　120行　130行　140行　150行　160行　170行

コンピュータでは、小節に区切る必要はないから、自分で作りやすいように切ってプログラムにすればいい。RUNさせて、うまく演奏されないときは、前の約束ごとをもう一度読み直そう。

●トルコ行進曲のプログラム

```
10 CLS
20 SCREEN 0,0
30 LOCATE 12,10:PRINT"*************
40 LOCATE 12,11:PRINT"*           *
50 LOCATE 12,12:PRINT"*  ﾄﾙｺ  ﾏｰﾁ  *
60 LOCATE 12,13:PRINT"*           *
70 LOCATE 12,14:PRINT"*************
80 PLAY "T120V8","T120V8","T120V8"
90 FOR I=0 TO 1
100 PLAY "O5L16BAG#A","R4","R4"
110 PLAY "C8R8","O3L16AR16O4CR16",
          "O4L16ER16O4ER16"
120 PLAY "O5L16DCO4BO5C","O4L16CR16CR16",
          "O4L16ER16O4ER16"
130 PLAY "O5E8R8","O3L16AR16O4CR16",
         "O3L16AR16O4ER16"
140 PLAY "O5L16FED#E","O4L16CR16CR16",
         "O4L16ER16ER16"
150 PLAY "O5L16BAG#A","O3L16AR16O4CR16",
         "O3L16AR16O4ER16"
160 PLAY "O5L16BAG#A","O3L16AR16O4CR16",
         "O3L16AR16O4ER16"
170 PLAY "O6L4C","O3L16AR16O4CR16",
         "O3L16AR16O4CR16"
180 PLAY "O5L8AO6C","O4L16CR16CR16",
         "O4L16ER16ER16"
190 PLAY "O5L8BAGA","O5L8BF#EF#",
         "O3L16ER16AR16AR16AR16"
200 PLAY "O5L8BAGA","O5L8BF#EF#",
         "O3L16ER16AR16AR16AR16"
210 PLAY "O5L8BAGF#E4","O5L8BF#ED#E4",
         "O3L16ER16AR16O2AR16O3AR16E4"
220 NEXT I
230 FOR I=0 TO 1
240 PLAY "O5L8EF","O5L8CD","R4"
250 PLAY "O5L16GR16GR16AGFE","O5L16ER16ER16R4",
         "O3L8CO4CO3EO4E"
260 PLAY "O5L4DE8F8","O4L8AGO5CD","O3GR4"
270 PLAY "O5L16GR16GR16AGFE","O5L16ER16ER16R4",
         "O3L8CO4CO3EO4E"
```

```
280 PLAY "O3L4DC8D8","O4L4BA8B8","O3G4R4"
290 PLAY "O5L16ER16ER16FEDC","O5L16CR16CR16R4",
         "O2L8AO3ACO4C"
300 PLAY "O4L4BO3C8D8","O4L8G#EAB","O3L4ER4"
310 PLAY "O5L16ER16ER16FEDC","O5L16CR16CR16R4",
         "O2L8AO3ACO4C"
320 PLAY "O4L4B","O4L4G#","O3E"
330 PLAY "L16BAG#A","R4","R4"
340 PLAY "O5C8R8","O3L16R16O4CR16",
         "O3L16AR16O4ER16"
350 PLAY "O5L16DCO4BO5C","O4L16CR16CR16",
         "O4L16ER16ER16"
360 PLAY "O5E8R8","O3L16AR16O4CR16",
         "O3L16AR16O4ER16"
370 PLAY "O5L16FED#E","O4L16CR16CR16",
         "O4L16ER16ER16"
380 PLAY "O5L16BAG#A","O3L16AR16O4CR16",
         "O3L16AR16O4ER16"
390 PLAY "O5L16BAG#A","O3L16AR16O4CR16",
         "O3L16AR16O4ER16"
400 PLAY "O6C4O5A8B8","O3L16FR16AR16AR16AR16",
         "O3L16FO4D#R16D#R16D#R16"
410 PLAY "O6L8CO5BAG#","O3L16ER16O4ER16O3DR16BR16",
         "O3L16ER16AR16DR16FR16"
420 PLAY "O5L8AEFD","O3L16CR16AR16DR16BR16",
         "O3L16CR16ER16DR16FR16"
430 PLAY "O5C4O4L8C.A32B32A4",
         "O3L16AR16AR16G#R16G#R16A4",
         "O3L16ER16ER16ER16ER16O2A4"
440 NEXT I
```

楽しいサウンド作り

数値を変えてためしてみよう

SOUND命令●サウンドを発生させる

サウンドを作るのが、SOUND命令だ。この命令を使うと、PLAY命令で作った音とはちがった、もっと細かい音まで出せるんだ。

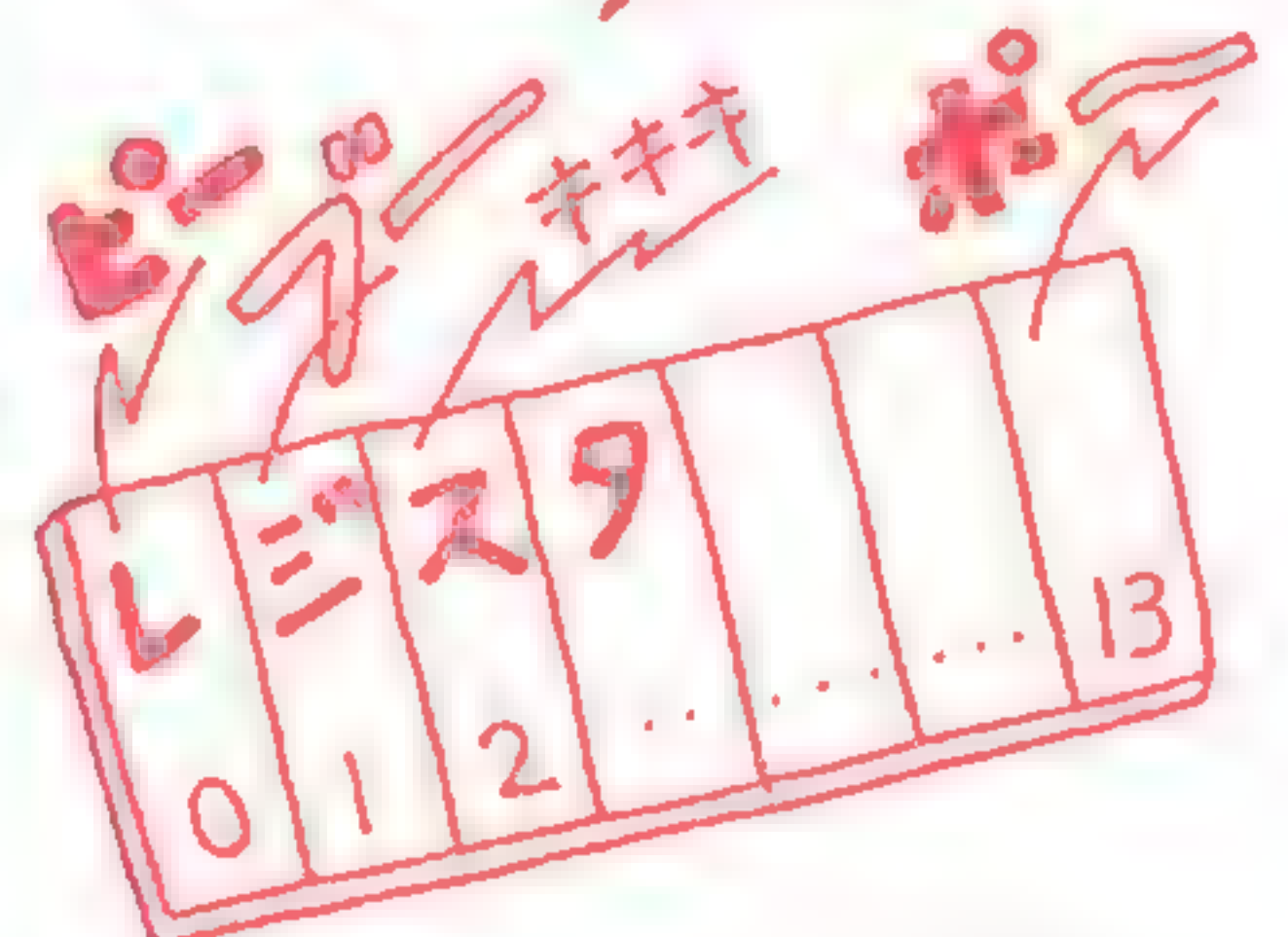
ただ、この機能の説明は、初心者の人にとってはかなり複雑でむずかしいのだ。

そこで、サウンドの書き方をきみに教えるから、あとは自分で数値を変えながらためしてみることだ。気に入った音が出たら、メモにとっておいて、ゲームに使うのもいい。

ゲームをおもしろくするのも、つまらなくするのもサウンドしだい。シーンとしたゲームなんて、スリルも迫力も出ない

SOUND　レジスタ番号，データ

レジスタというのは、音を発生させるところで、0から13まで、ぜんぶで14の発生源がある。同じレジスタでも、与えるデータによって鳴る音が違うんだ。データには数値を入れるのだけど、0から13までのそれぞれのレジスタに入れられる数値の範囲は決まっている。次のページに、その範囲を書いておいたから、この範囲を超えないように注意しよう。

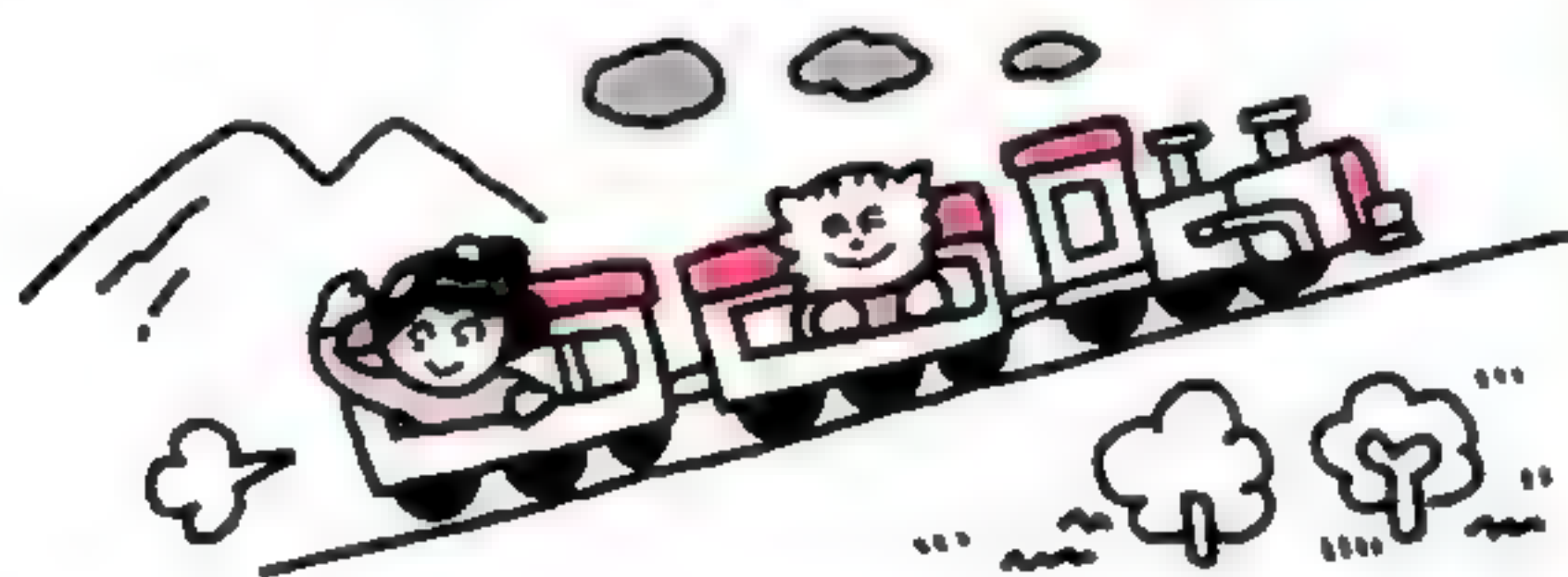
各レジスタには、それぞれいろいろな役わりがあるんだよ

レジスタ番号	データの範囲
0	0～255
1	0～15
2	0～255
3	0～15
4	0～255
5	0～15
6	0～31
7	0～63
8	0～16
9	0～16
10	0～16
11	0～255
12	0～255
13	0～15

レジスタは14こあるけど、ぜんぶのレジスタを使わなければいけないということはないよ。3つや4つのレジスタを使って作る音もあるんだ。汽車は7つのレジスタでできる。

サウンド「汽車」

```
10 SOUND  6,  13
20 SOUND  7,  35
30 SOUND  8,  16
40 SOUND  9,  16
50 SOUND 11, 100
60 SOUND 12,   1
70 SOUND 13,  10
```

左がレジスタの番号、右がその数値だ。同じ汽車でも、使うレジスタやデータの数値を変えると、坂を登っている音がでるよ。

サウンド「登り坂の汽車」

```
10 SOUND  0, 200
20 SOUND  7,  35
30 SOUND  8,  16
40 SOUND  9,  16
50 SOUND 11, 100
60 SOUND 12,   4
70 SOUND 13,  14
```

サウンド作りプログラム

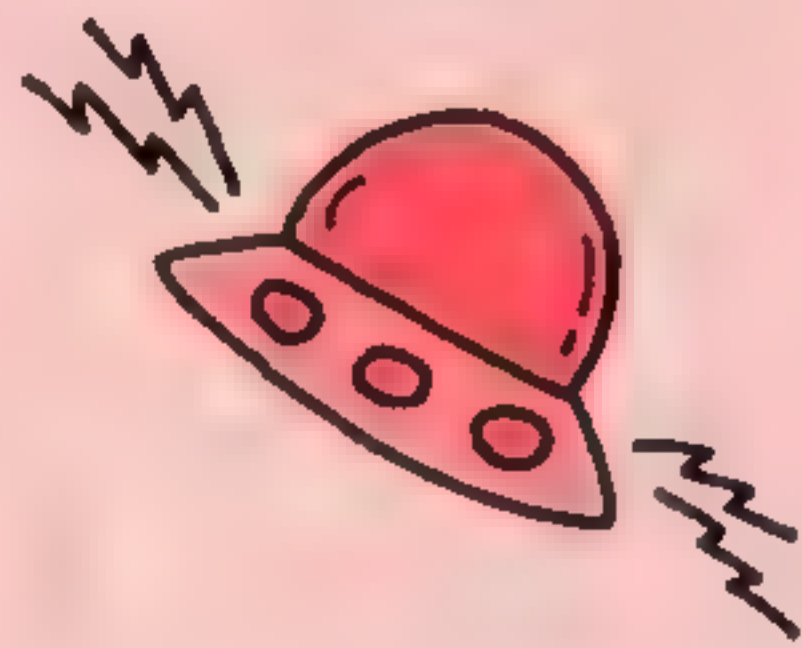
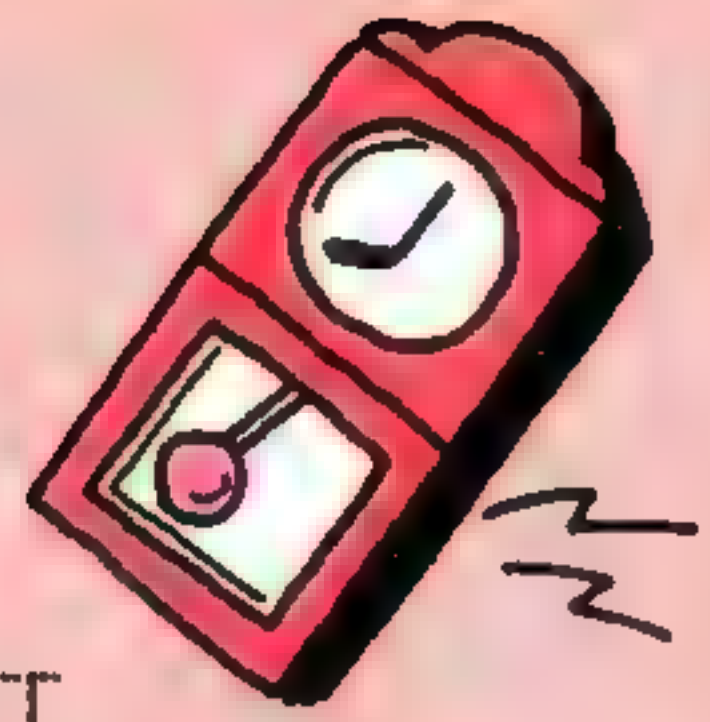
最後に、数値を入力するだけで、いろいろなサウンドが飛び出すプログラムをのせておこう。

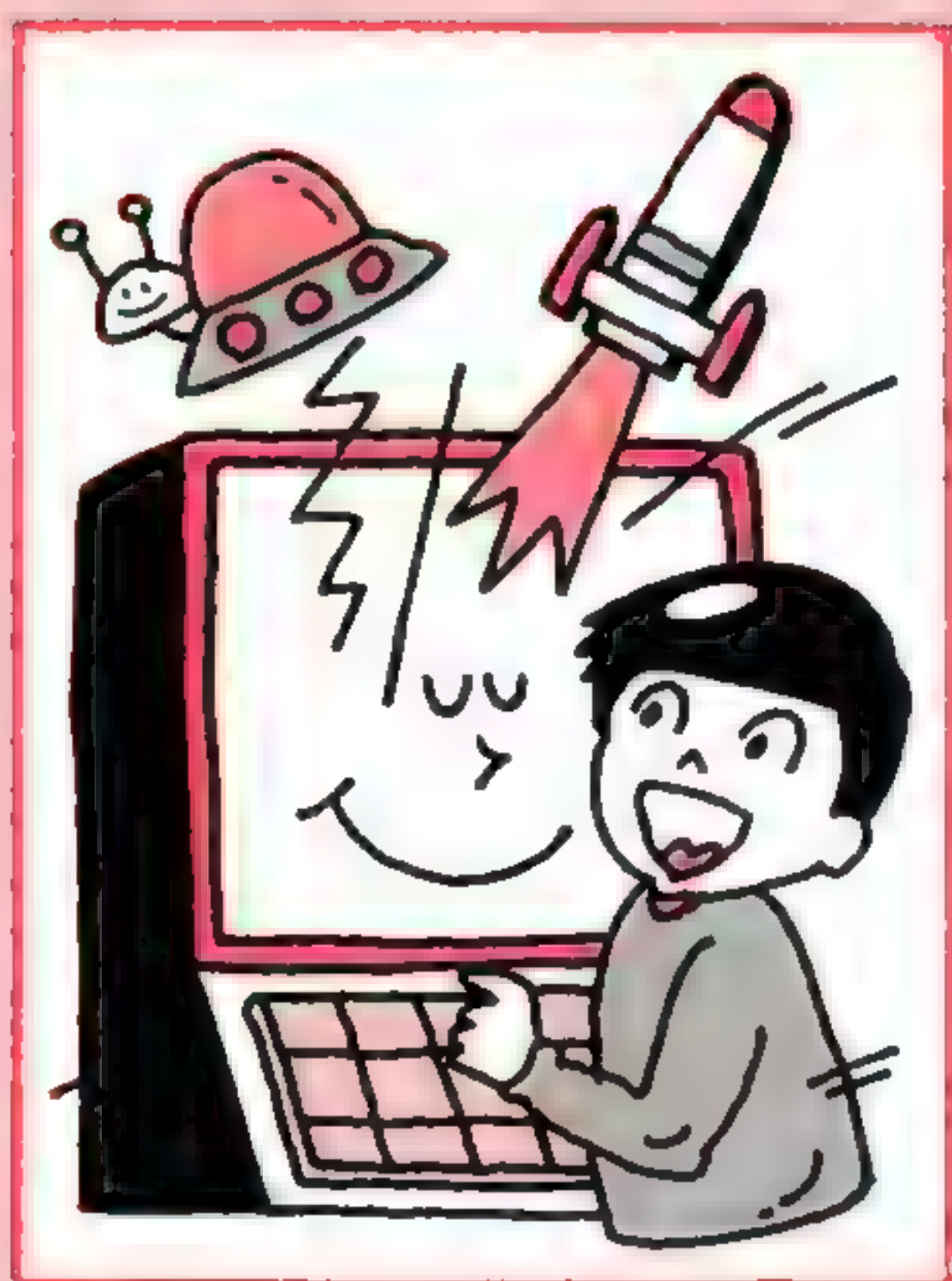
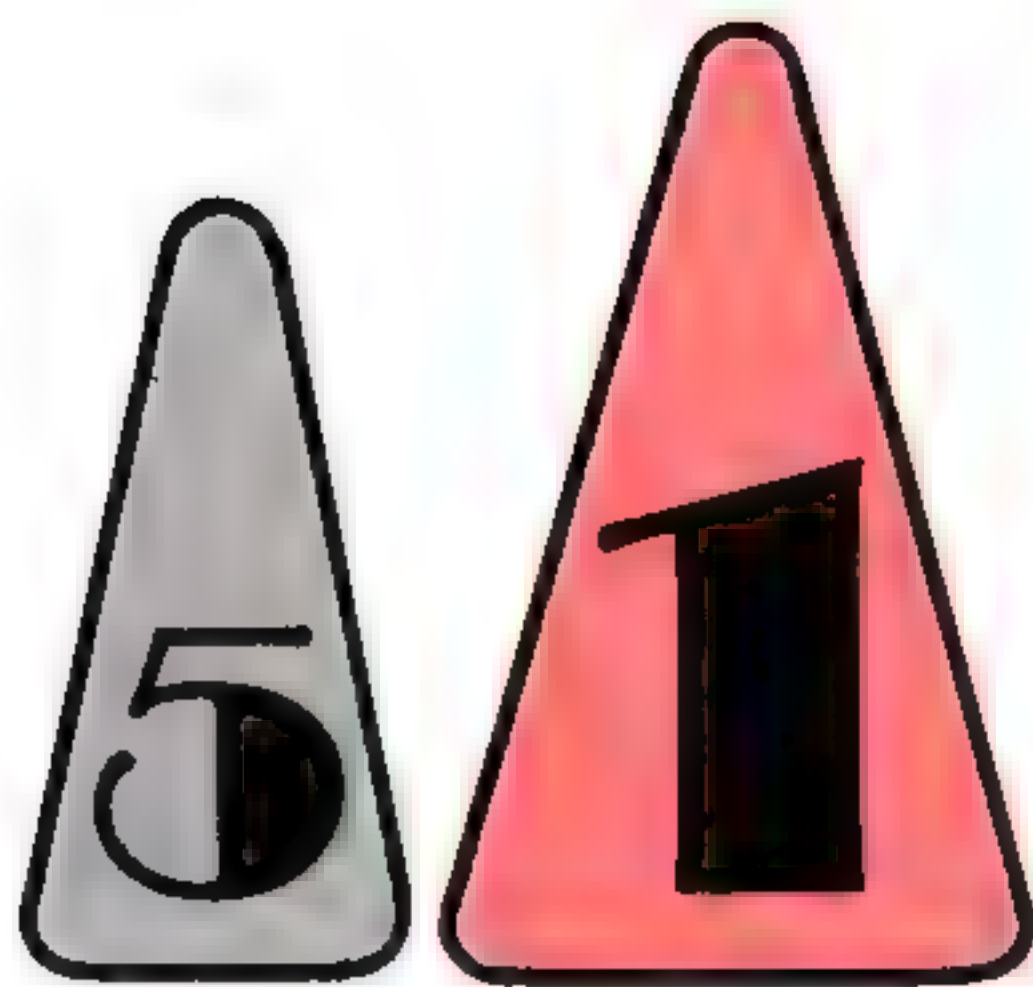
プログラムの80行から210行までに各レジスタ番号とその役わりが書いてあるね。ここには、音の鳴らし方や音の大きさなどいろいろな機能がある。とにかくどんどん数値を入れてみよう。

```
10  SOUND   1,  3
20  SOUND   8, 16
30  SOUND  12, 10
40  SOUND  13,  8
```

ドラムの音のプログラムはこうだ。ためしに入れてごらん

```
10 CLS
20 KEY OFF
30 DIM A(13)
40 PRINT"        ********************"
50 PRINT"        *   SOUND EDITER   *"
60 PRINT"        ********************"
70 PRINT"SOUND No."
80 PRINT" 0....Aﾁｬﾝﾈﾙ(0-255)    :"
90 PRINT" 1....Aﾁｬﾝﾈﾙ(0-15)     :"
100 PRINT" 2....Bﾁｬﾝﾈﾙ(0-255)    :"
110 PRINT" 3....Bﾁｬﾝﾈﾙ(0-15)     :"
120 PRINT" 4....Cﾁｬﾝﾈﾙ(0-255)    :"
130 PRINT" 5....Cﾁｬﾝﾈﾙ(0-15)     :"
140 PRINT" 6....ﾉｲｽﾞ(0-31)       :"
150 PRINT" 7....ｵﾝｹﾞﾝ(0-63)      :"
160 PRINT" 8....Aｵﾝﾘｮｳ(0-16)     :"
170 PRINT" 9....Bｵﾝﾘｮｳ(0-16)     :"
180 PRINT"10....Cｵﾝﾘｮｳ(0-16)     :"
190 PRINT"11....ｴﾝﾍﾞﾛｰﾌﾟ(0-255)  :"
200 PRINT"12....ｴﾝﾍﾞﾛｰﾌﾟ(0-255)  :"
210 PRINT"13....ｴｰﾊﾟﾀｰﾝ(0-15)    :"
220 PRINT"-----------------------------------"
```

```
230 FOR N=0TO13
240 LOCATE22,4+N:INPUTB
250 F=1:GOSUB450
260 A(N)=B
270 NEXT
280 FORN=0TO13:SOUND N,A(N):NEXT
290 LOCATE 5,20
300 PRINT"ｼｭｳｾｲ(No,ｽｳﾁ)"
310 LOCATE 18,20:INPUTC,D
320 N=C:B=D:F=2:GOSUB450
330 A(C)=D
340 LOCATE23,4+C:PRINTD
350 GOSUB370
360 GOTO 290
370 FORN=0TO13:SOUND N,A(N):NEXT:RETURN
380 FORN=0TO13:SOUND N,A(N):NEXT:RETURN
390 FORM=0TO6:LOCATE18,20
400 PRINT"DATA ERROR"
410 FORI=0TO 50:NEXT:BEEP
420 LOCATE18,20
430 PRINT"              "
440 NEXT:RETURN
450 IFN=0THENIFB>255THENGOSUB390:H=1
460 IFN=1THENIFB>15THENGOSUB390:H=1
470 IFN=2THENIFB>255THENGOSUB390:H=1
480 IFN=3THENIFB>15THENGOSUB390:H=1
490 IFN=4THENIFB>255THENGOSUB390:H=1
500 IFN=5THENIFB>15THENGOSUB390:H=1
510 IFN=6THENIFB>31THENGOSUB390:H=1
520 IFN=7THENIFB>63THENGOSUB390:H=1
530 IFN=8THENIFB>16THENGOSUB390:H=1
540 IFN=9THENIFB>16THENGOSUB390:H=1
550 IFN=10THENIFB>16THENGOSUB390:H=1
560 IFN=11THENIFB>255THENGOSUB390:H=1
570 IFN=12THENIFB>255THENGOSUB390:H=1
580 IFN=13THENIFB>15THENGOSUB390:H=1
590 IFF=1THENIFH=1THENF=0:H=0:RETURN240
600 IFF=2THENIFH=1THENF=0:H=0:RETURN310
610 F=0:RETURN
```

パート5

ゲームで遊ぼう

8×8のキャラクタを動かす

ロケットを画面に表示させる

スプライトパターン●ゲームのコマ作り

パソコンゲームのスプライト機能は、作ったキャラクタを画面上に表示して、自由自在に動かすことができる、とても便利な機能なんだ。パソコンゲームには、絶対に欠かせないものだよ

パソコンゲームやテレビゲームには、UFOやインベーダー、ロケットやキャノン砲などのキャラクタがたくさん飛び出してくる。

これらのキャラクタは、いったいどうやって作られているのかなあ、そして、どうすれば、あんなふうに動かすことができるのだろう。

このキャラクタをスプライトパターンというんだ。キャラクタを動かす命令がPUT SPRITE命令というんだ。

まず、キャラクタの作り方から説明しよう。キャラクタは8×8と16×16の2つのタイプがある。16×16の方が、より細かな図柄を表現できるんだ。

はじめに、8×8のかんたんなキャラクタを作ってみよう。

紙に、縦8個横8個の正方形のマスを書く。マス目の左側に、縦に1から8までの番号をふっておく。そして、このマス目を好きなように黒く塗りつぶして、何か形のある図形を作ってごらん。

●8×8のスプライトパターンを定義する

1 ……………&B00001000
2 ……………&B00001000
3 ……………&B00011100
4 ……………&B00011100
5 ……………&B01011101
6 ……………&B01111111
7 ……………&B01111111
8 ……………&B01011101

さて、どんなふうに塗りつぶしたのかをコンピュータに教えてやらなければ、画面には表示できない。

これを「スプライトパターンを定義する」という。

この伝え方は、2進数、10進数、16進数の3通りある。どれを使っても同じ形が表示できるよ。

ここでは、いちばんわかりやすい2進数で説明しよう。

2進数であらわすときは、マス目の白い部分を0、黒く塗りつぶしたところを1にして、スプライトパターンを定義するんだ。

そして、その前に「2進数であらわしているんだよ」というサインの&Bをつける。

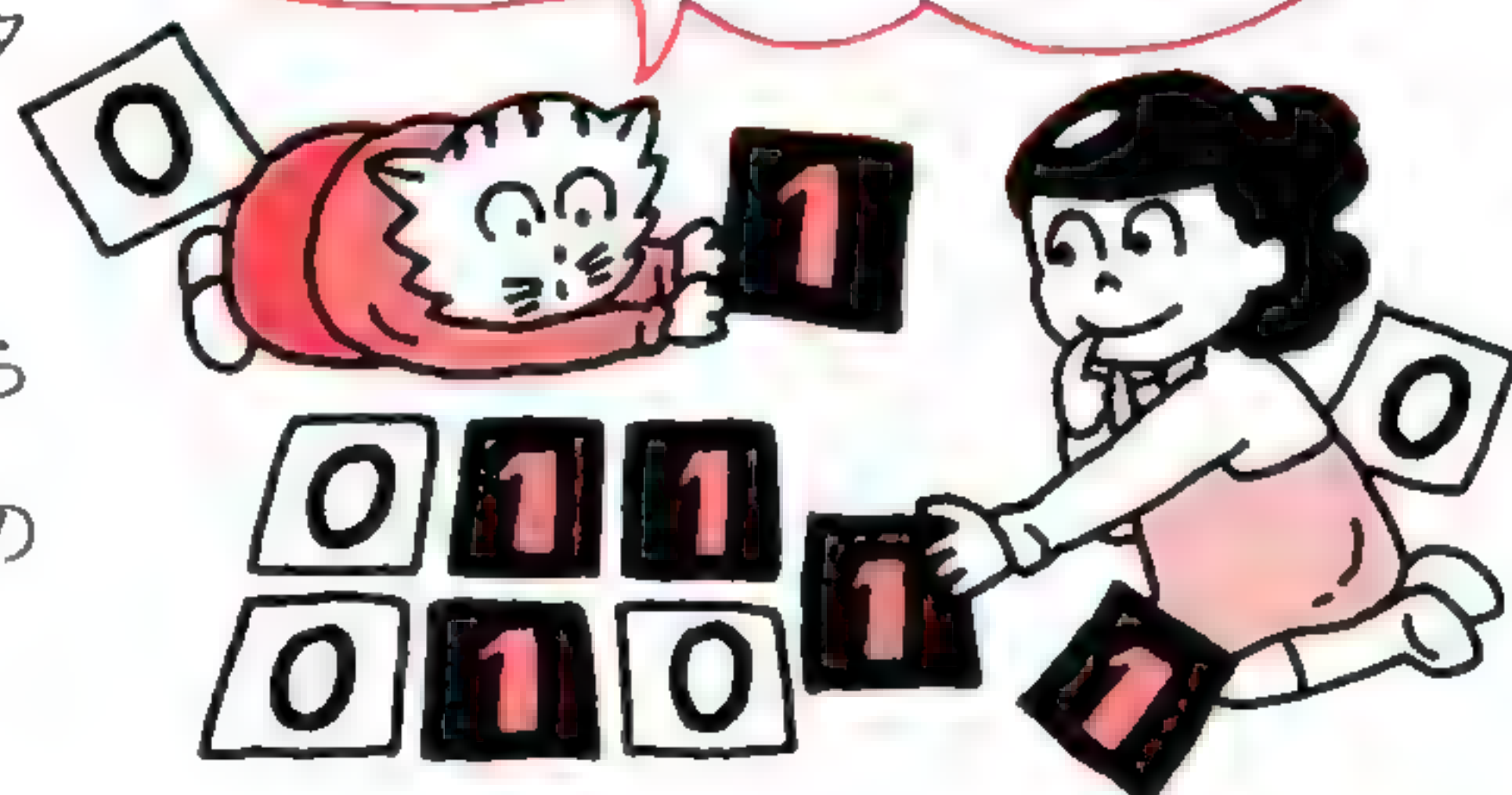
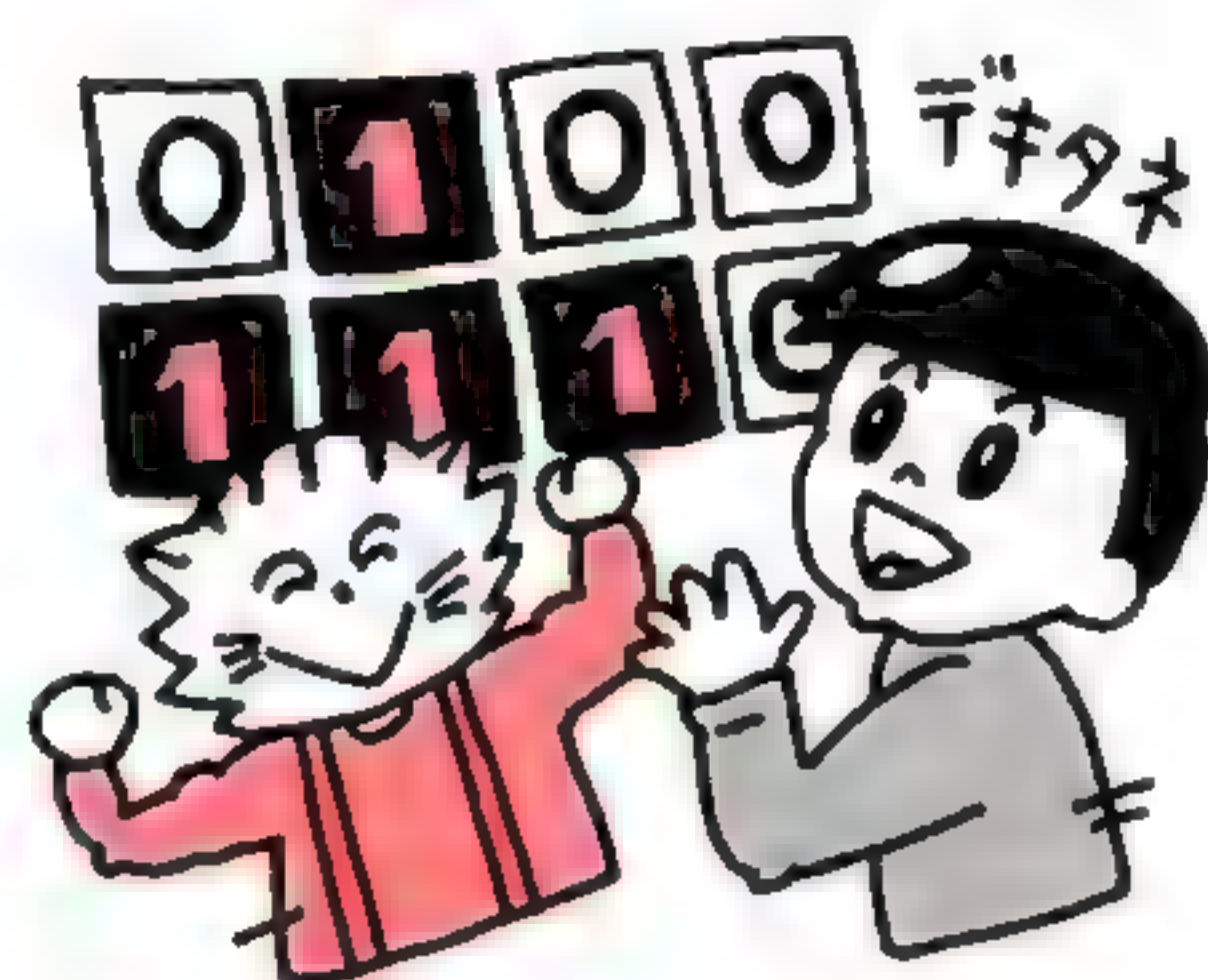
たとえば、5段目は

↓

&B01011101

こうしてマス目の黒と白を数値に置きかえることを、「キャラクタコードに直す」というんだ。書き方は、

CHR$（&B01011101）

となる

マス目の左側の数の意味がわかったかな？パターン全体を定義するのを、SPRITE＄というんだけど、Ｓ１＄、Ｓ２＄………は、そのスプライトパターンの１段目、２段目という意味だ。

パターン全体を定義するには、そう、Ｓ１＄からＳ８＄までをたせばいい。

SPRITE＄（0）＝S1＄＋S2＄＋S3＄＋S4＄＋S5＄＋S6＄＋S7＄＋S8＄

さぁ、プログラムに直してみよう。

```
10 SCREEN 1,0
20 S1$=CHR$(&B00001000)
30 S2$=CHR$(&B00001000)
40 S3$=CHR$(&B00011100)
50 S4$=CHR$(&B00011100)
60 S5$=CHR$(&B01011101)
70 S6$=CHR$(&B01111111)
80 S7$=CHR$(&B01111111)
90 S8$=CHR$(&B01011101)
100 SPRITE$(0)=S1$+S2$+S3$+S4$+S5$+S6$+S7$+S8$
```

スプライトパターンを表示する

スプライトパターンを画面に表示させるには、いくつかのきまりがあるから、おぼえておこう。

スプライトを表示するときのSCREEN

SCREEN　0は使えない。あとは、テキスト画面でもグラフィック画面でもOK。だから、SCREEN　1または2または3だ。

スプライトの大きさ

0………8×8ドットで文字の大きさと同じ

1………8×8ドットで、縦横の長さが、ふつうの文字の2倍

2………16×16ドット

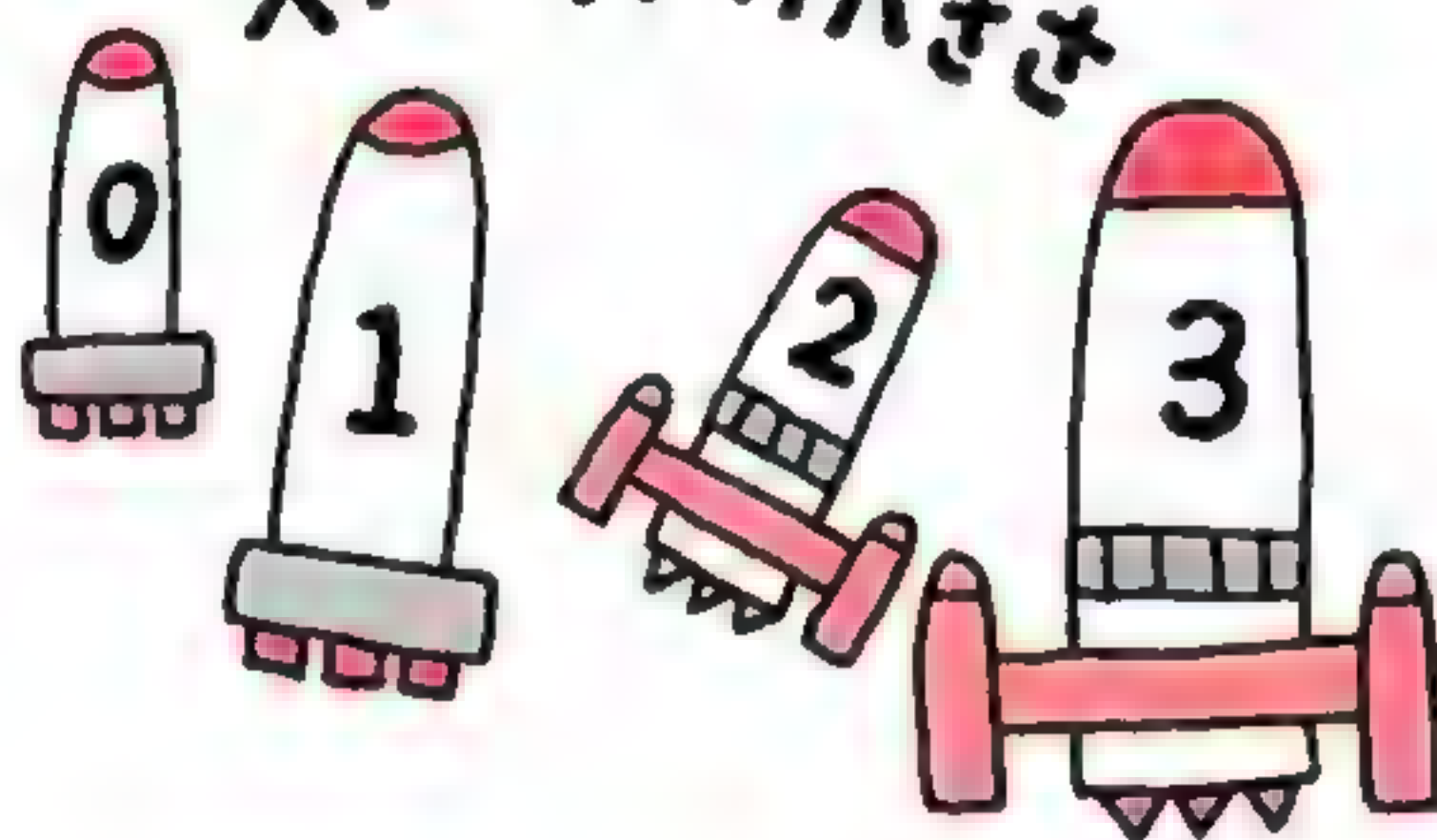
3………16×16ドットで、縦横の長さがそれぞれ2倍

スプライトのパターン番号

スプライトパターンをいくつも使うときは、どのパターンをどこに定義したのか、番号をつけて区別してやらなければならない。これがスプライト番号で、左ページ

SPRITE$（0）＝

の0がその番号だ。スプライト番号は、0～255の範囲で設定し、画面に最高256個のパターンが用意できるよ。

このロケットのパターンはスプライト番号0に定義されたんだね

PUT SPRITE命令●スプライトパターンを動かす

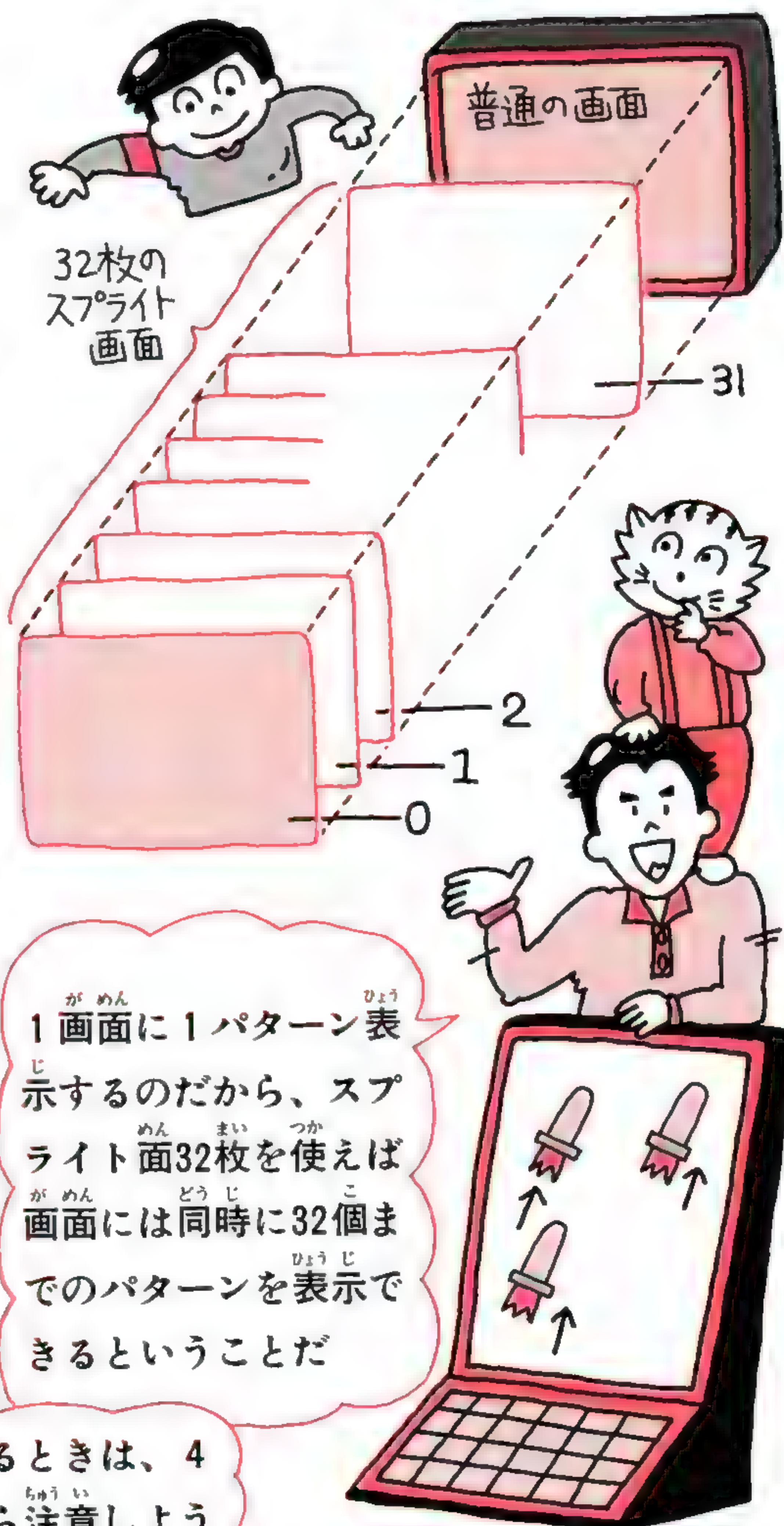
スプライトパターンを画面に表示させる命令が、PUT SPRITE命令だ。命令文の書き方は、こうだ。

PUT SPRITE スプライト面番号,(表示する位置),表示する色,スプライト番号

●スプライト面番号

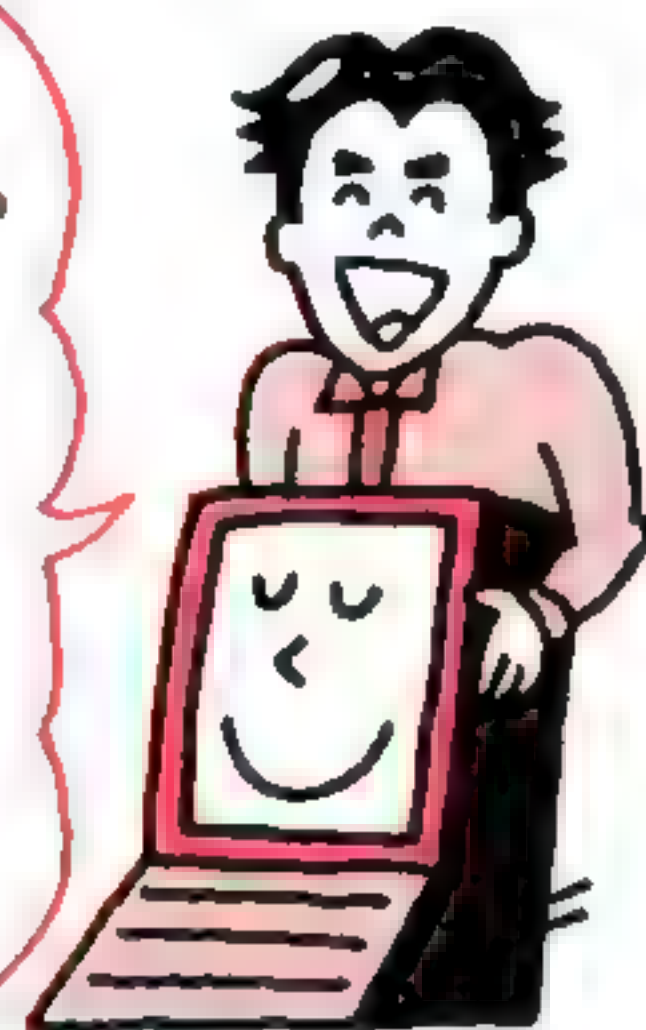
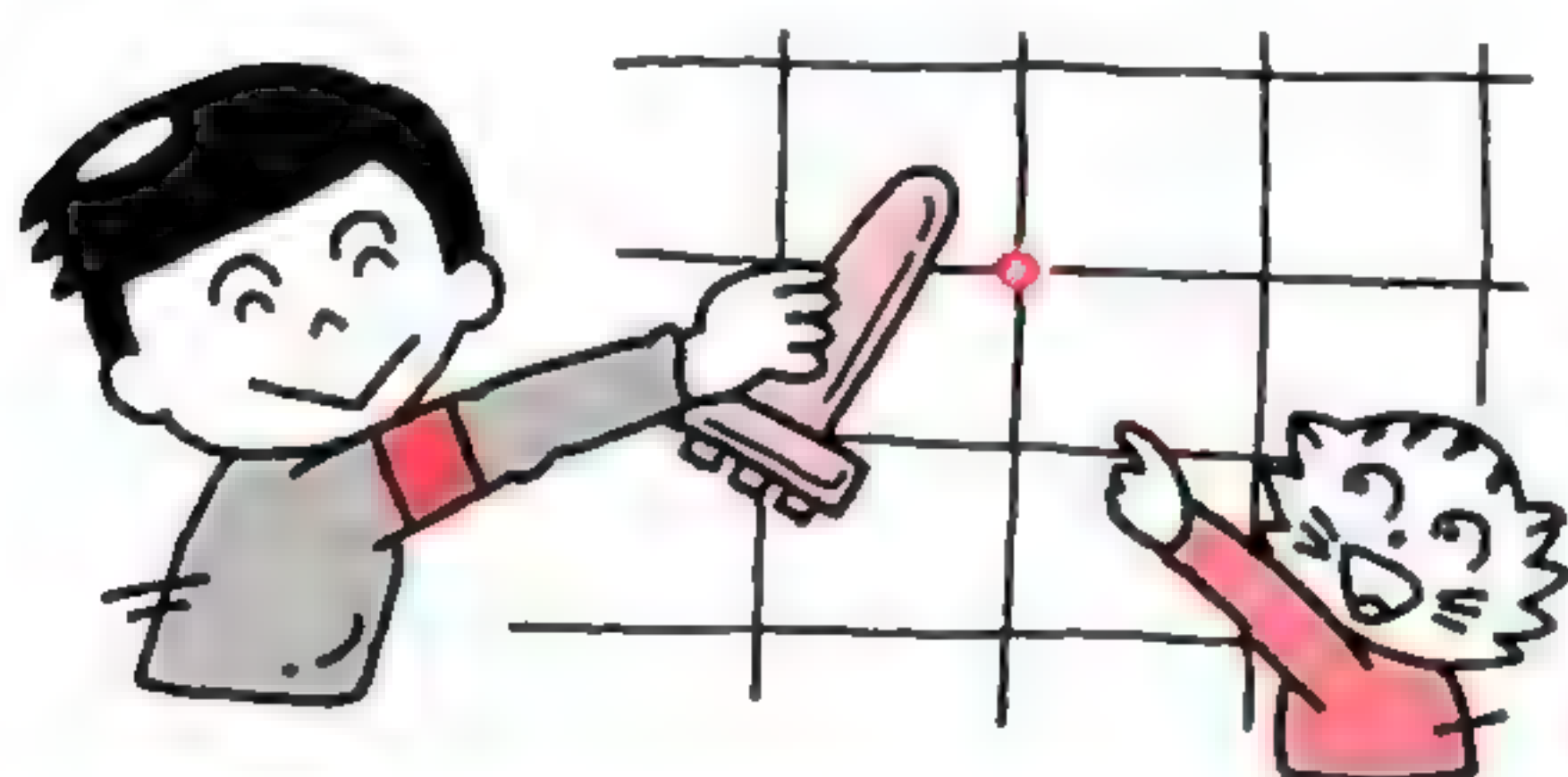
ふつうの画面とは別に、スプライトパターンだけを表示するための専用画面で、0から31まで、ぜんぶで32枚用意されている。

1つのスプライト画面には、1つのスプライトしか表示できない。

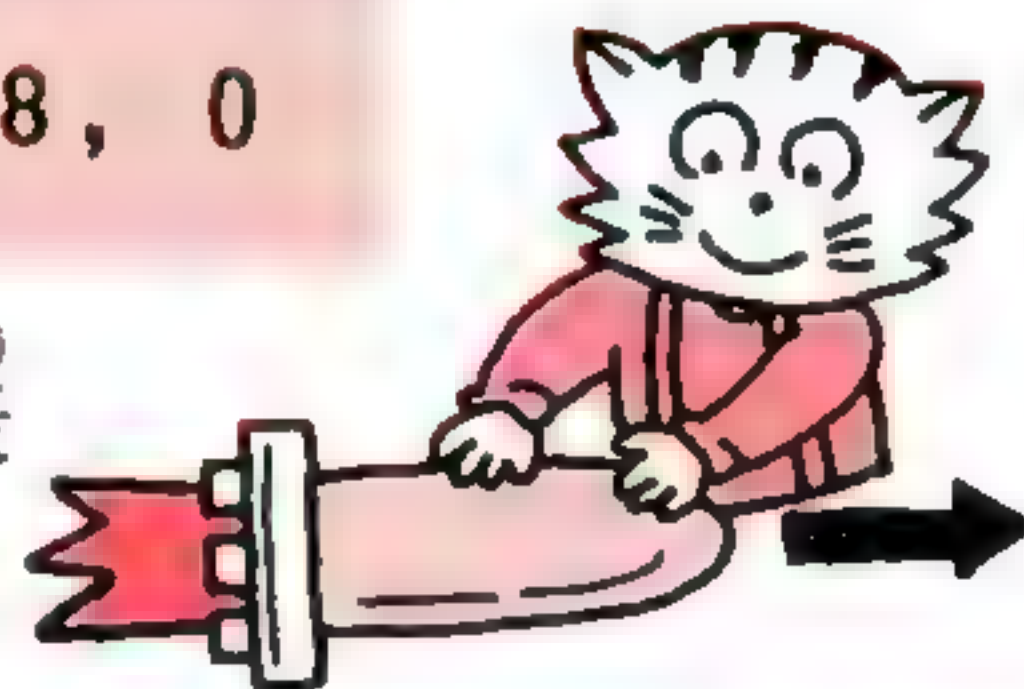
同じスプライト画面の別の位置にパターンを表示すると、その瞬間、前のパターンは姿を消してしまう。これを利用すれば、パターンが一瞬のうちに移動したり、変身したかのように表現できるぞ。

ただし、横に並べて表示するときは、4個までと決められているから注意しよう

表示する位置

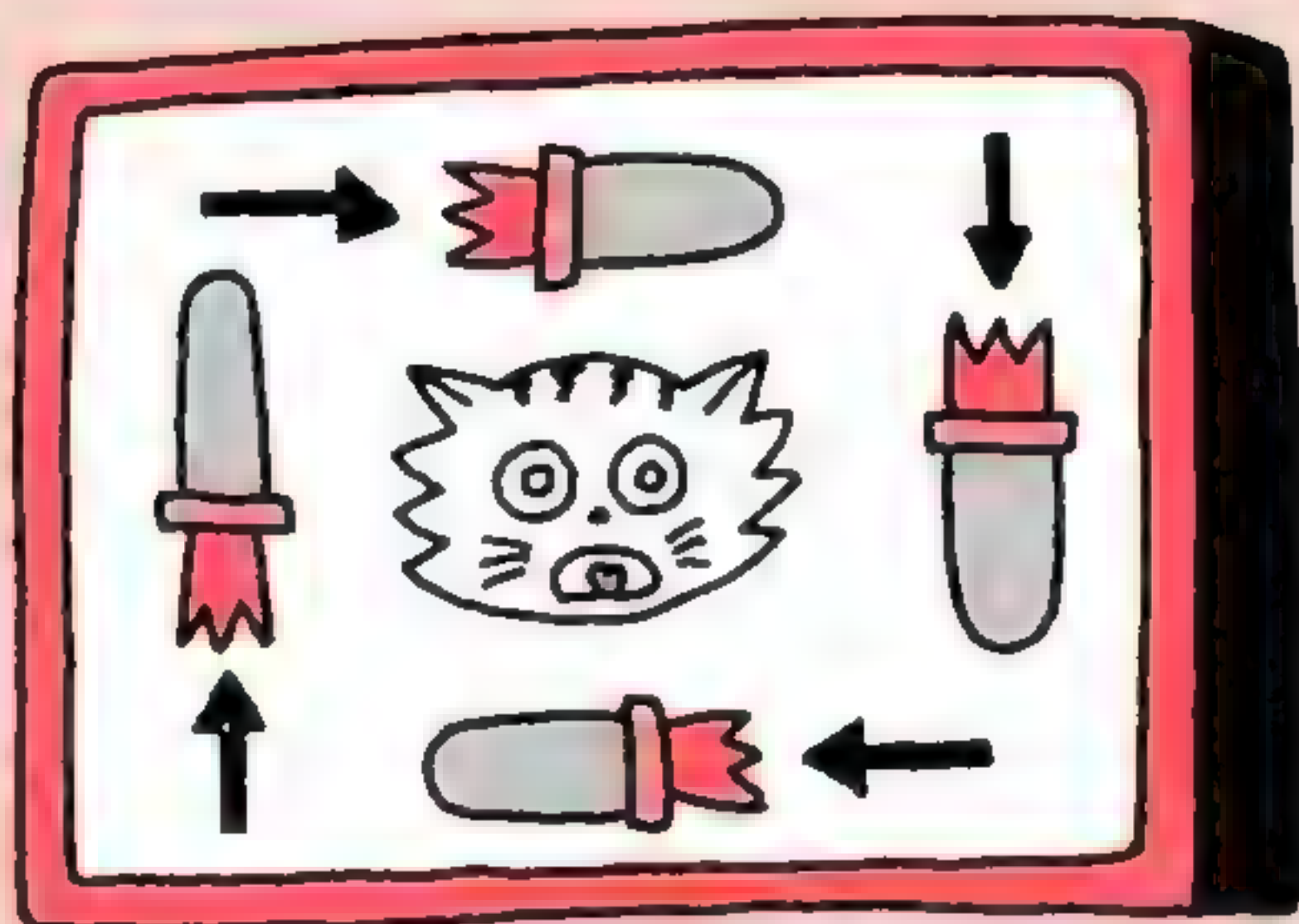
画面に表示する位置は、（X座標，Y座標）のことだ。X（横の座標）は、−32〜255の範囲で、Y（縦の座標）は−32〜191の範囲で指定するんだ。

さあ、画面に表示させてみよう。次のプログラムを追加して、RENUMでそろえてから、実行してごらん。まず、使うSCREENと大きさを指定してから

```
9 SCREEN 1, 0
```

次に、いよいよPUT SPRITEで画面に表示させるぞ。座標（0，100）に、赤（8）で表示させてみよう

```
110 PUT SPRITE 0, (0, 100), 8, 0
```

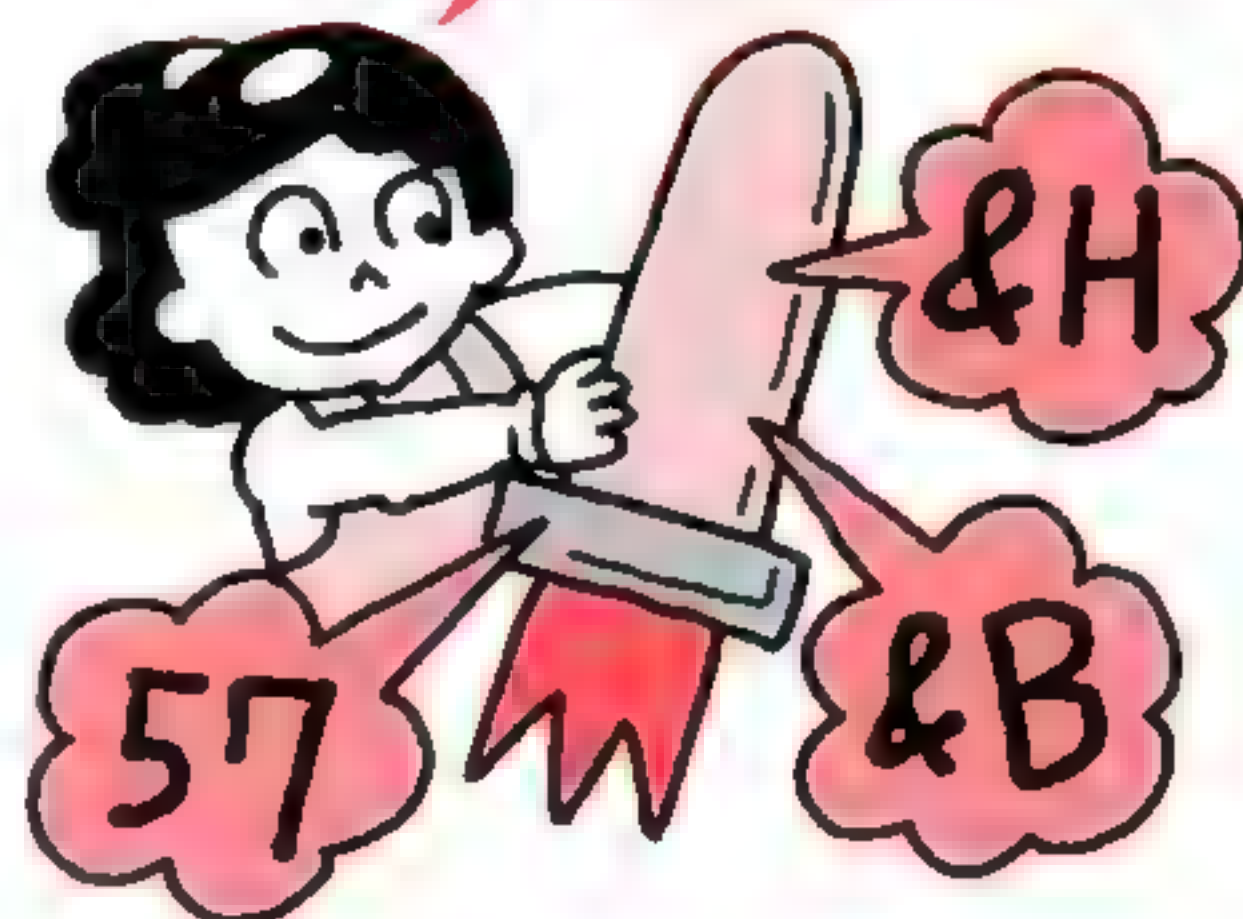
画面に出ただけじゃつまらない。スプライトの表示場所を変えれば、左から右に動くようになるぞ。

```
10 SCREEN 1,0
20 S1$=CHR$(&B00001000)
30 S2$=CHR$(&B00001000)
40 S3$=CHR$(&B00011100)
50 S4$=CHR$(&B00011100)
60 S5$=CHR$(&B01011101)
70 S6$=CHR$(&B01111111)
80 S7$=CHR$(&B01111111)
90 S8$=CHR$(&B01011101)
100 SPRITE$(0)=S1$+S2$+S3$+S4$+S5$+S6$+S7$+S8$
110 PUT SPRITE 0,(0,100),8,0
120 FOR N=1 TO 230 STEP 2
130 PUT SPRITE 0,(N,100),8,0
140 NEXT
150 COLOR 15,4,7
```

ロケットを上下左右に動かそう

```
10 SCREEN 1,0:KEY OFF
20 FOR I=0 TO3
30 SP$=""
40 FOR J=0 TO7
50 READ A:A$=CHR$(A)
60 SP$=SP$+A$
70 NEXT J
80 SPRITE$(I)=SP$
90 NEXT I
100 X=120:Y=88
110 A=STICK(0)
120 IF A=1 AND Y<>0 THEN Y=Y-8:SP=0
130 IF A=3 AND X<>240 THEN X=X+8:SP=1
140 IF A=5 AND Y<>184 THEN Y=Y+8:SP=3
150 IF A=7 AND X<>8 THEN X=X-8:SP=2
160 PUT SPRITE 0,(X,Y),8,SP
170 GOTO 110
180 DATA &H8,&H8,&H1C,&H1C,&H5D,&H7F,&H7F,&H5D
190 DATA &HF0,&H60,&HFC,&HFF,&HFC,&H60,&HF0,0
200 DATA &HF,&H6,&H3F,&HFF,&H3F,&H6,&HF,0
210 DATA &H5D,&H7F,&H7F,&H5D,&H1C,&H1C,&H8,&H8
```

カーソルキーを押した方向に、ロケットが動くプログラムだ。

まず、画面の中央にスプライトで表示させたロケットがあらわれる。そのあと、上下左右のカーソルキーを押してみよう。方向によって、ロケットの向きも変わるよ。

このプログラムでは、16進数でスプライトパターンを定義してある。180行から210行のDATA文が、4つの向きのスプライトパターンだよ。

180行から210行のデータを、さっきの2進数に変えても同じパターンが出るよ

16×16のインベーダーだ

8×8が4つ集まったと考えよう

8×8ドットのパターンの表示では、縦に8個横に8個、計64個のマス目があった。

16×16ドットのパターンなら、縦に16個、横に16個の合計256個のマス目があることになる。これだけマス目が多ければ、かんたんな8×8のパターンよりも、もっとおもしろいパターンも表示できるぞ。

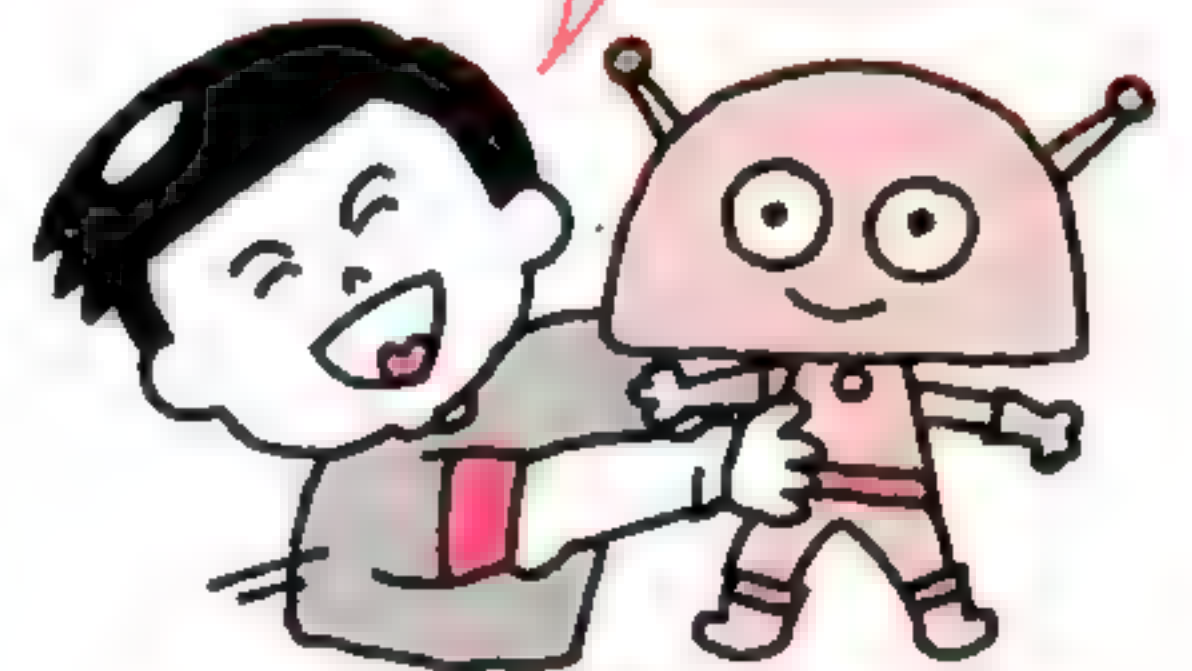
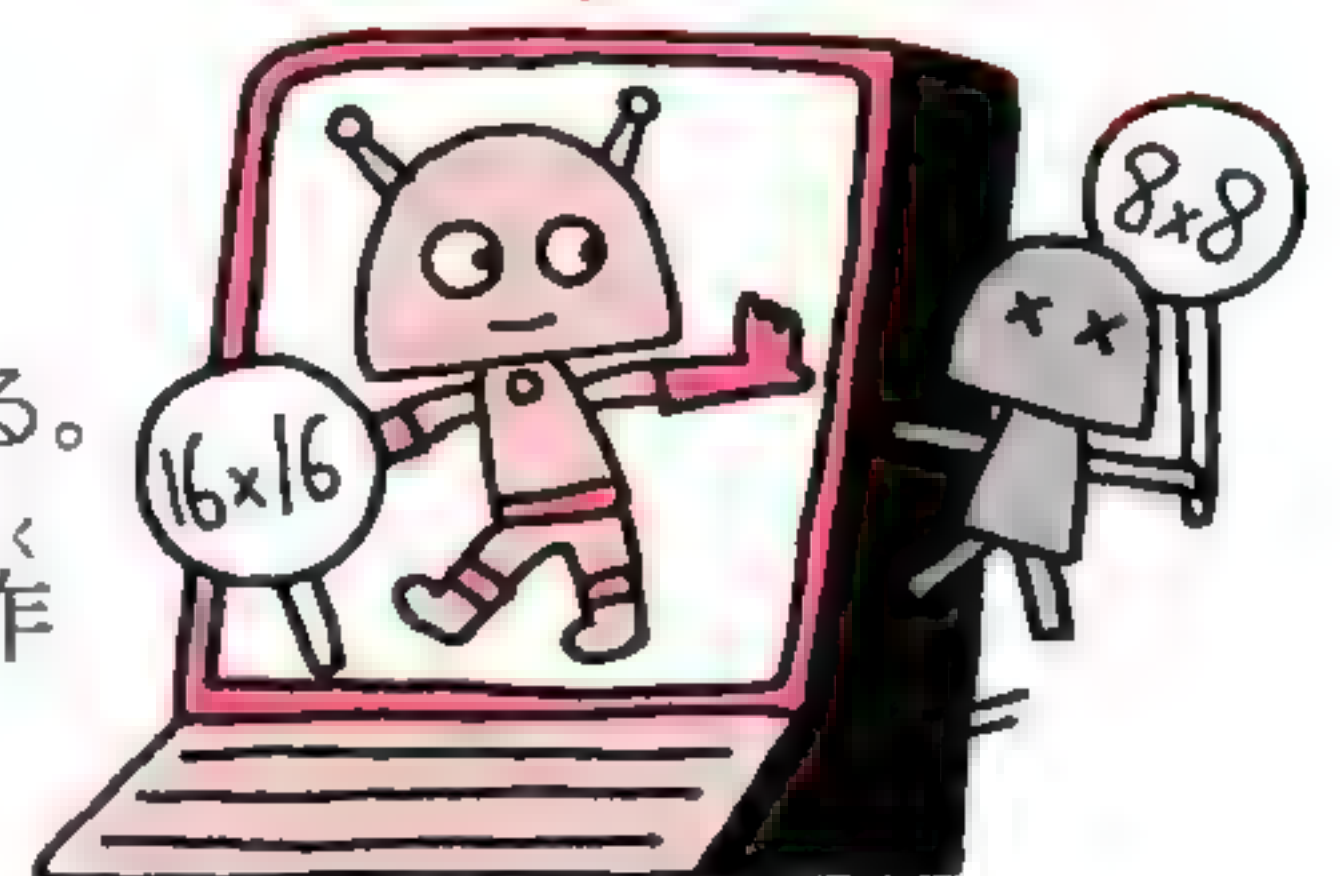
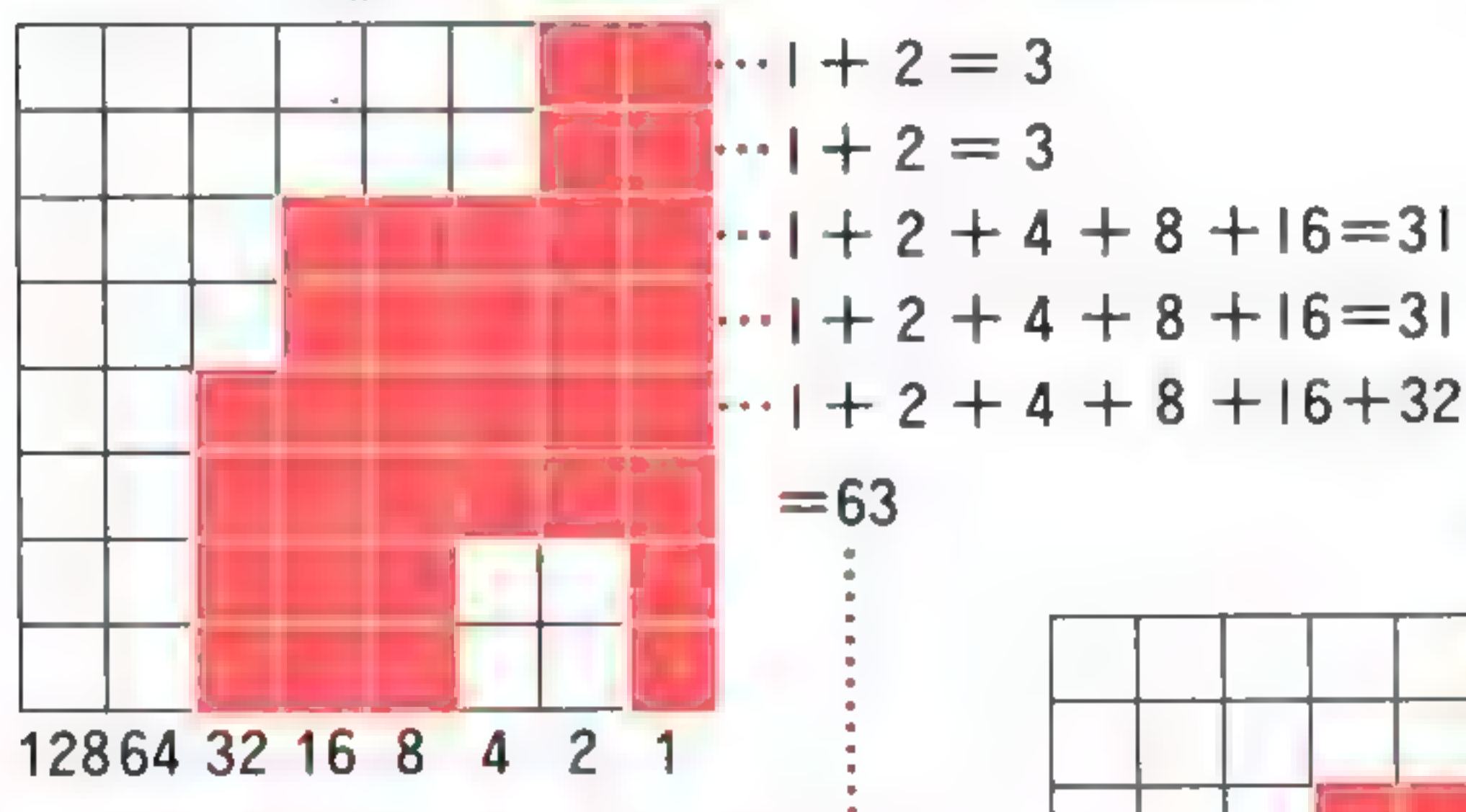
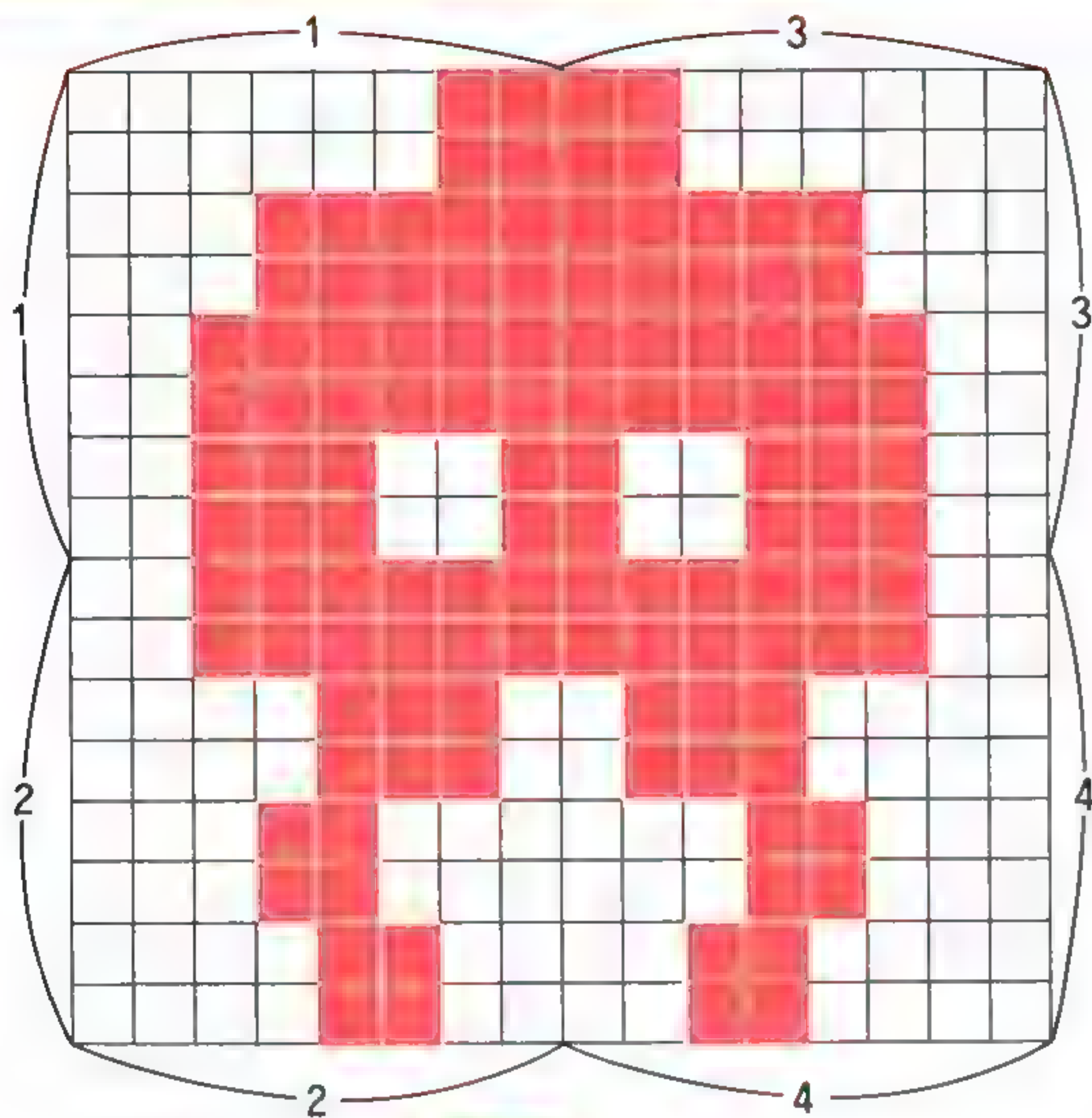
ただし、注意しなければならないことがある。16×16のパターンを表示させるときは、SCREEN　1は、使えないよ。SCREEN　2または3のグラフィック・モードを使わないと表示できないからね。

それともうひとつ、8×8のパターンは、最高256個まで用意できたけれど、16×16では、最高64個までだよ。つまり

SPRITE$（　）

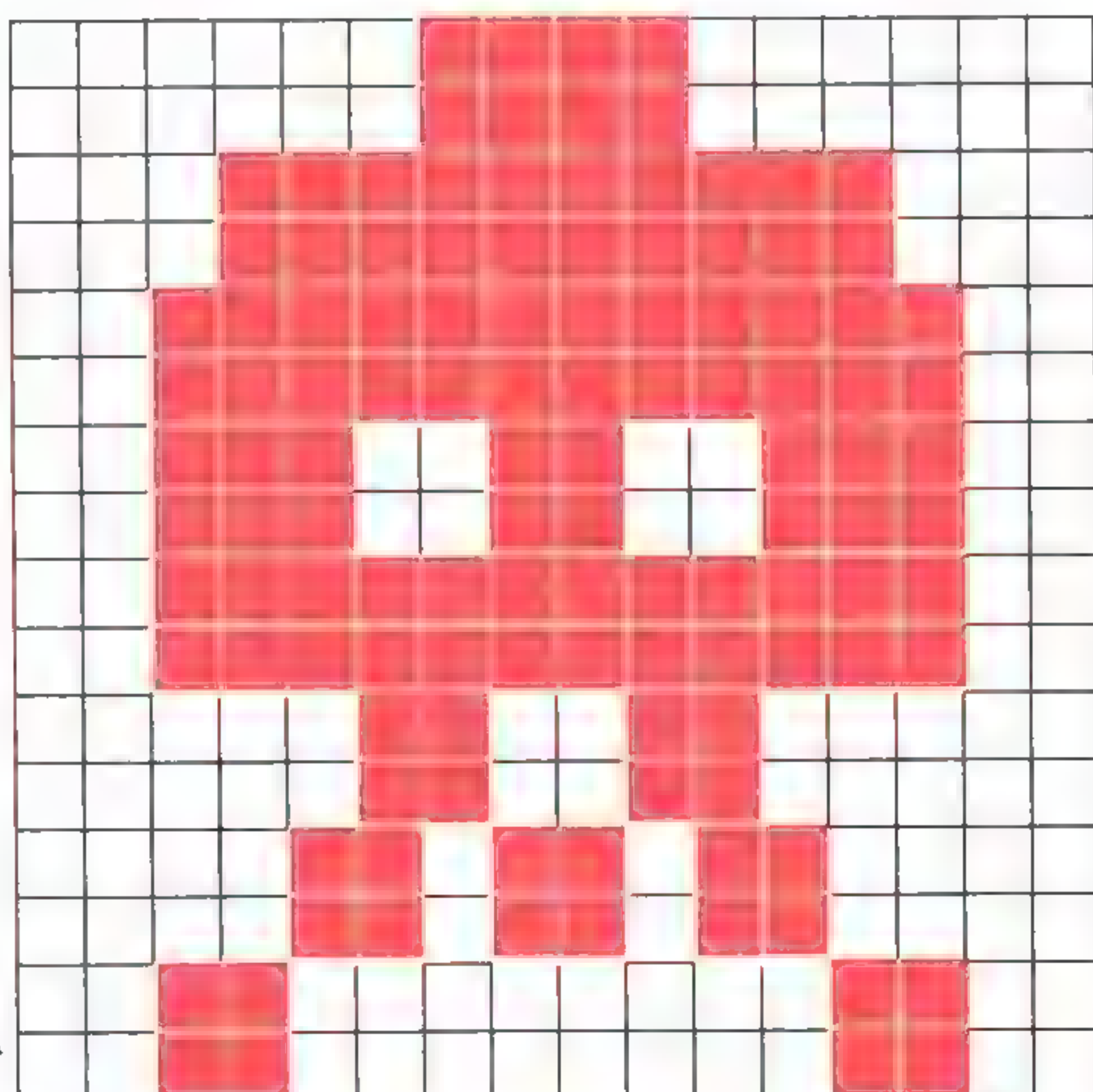
のカッコの中は、0から63までと決められている。

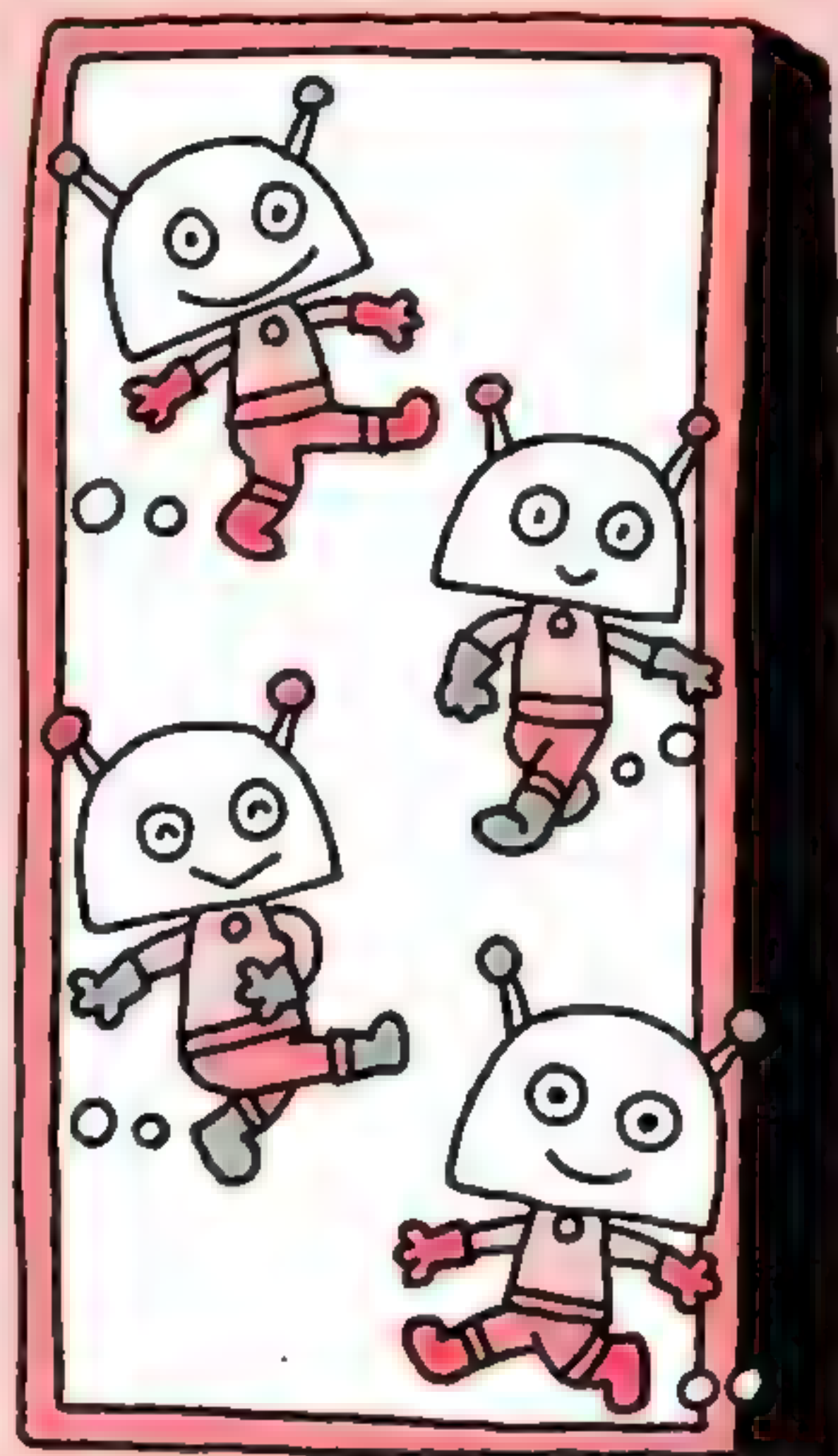
さあ、ここでは、インベーダーのような形を作ってみたぞ。

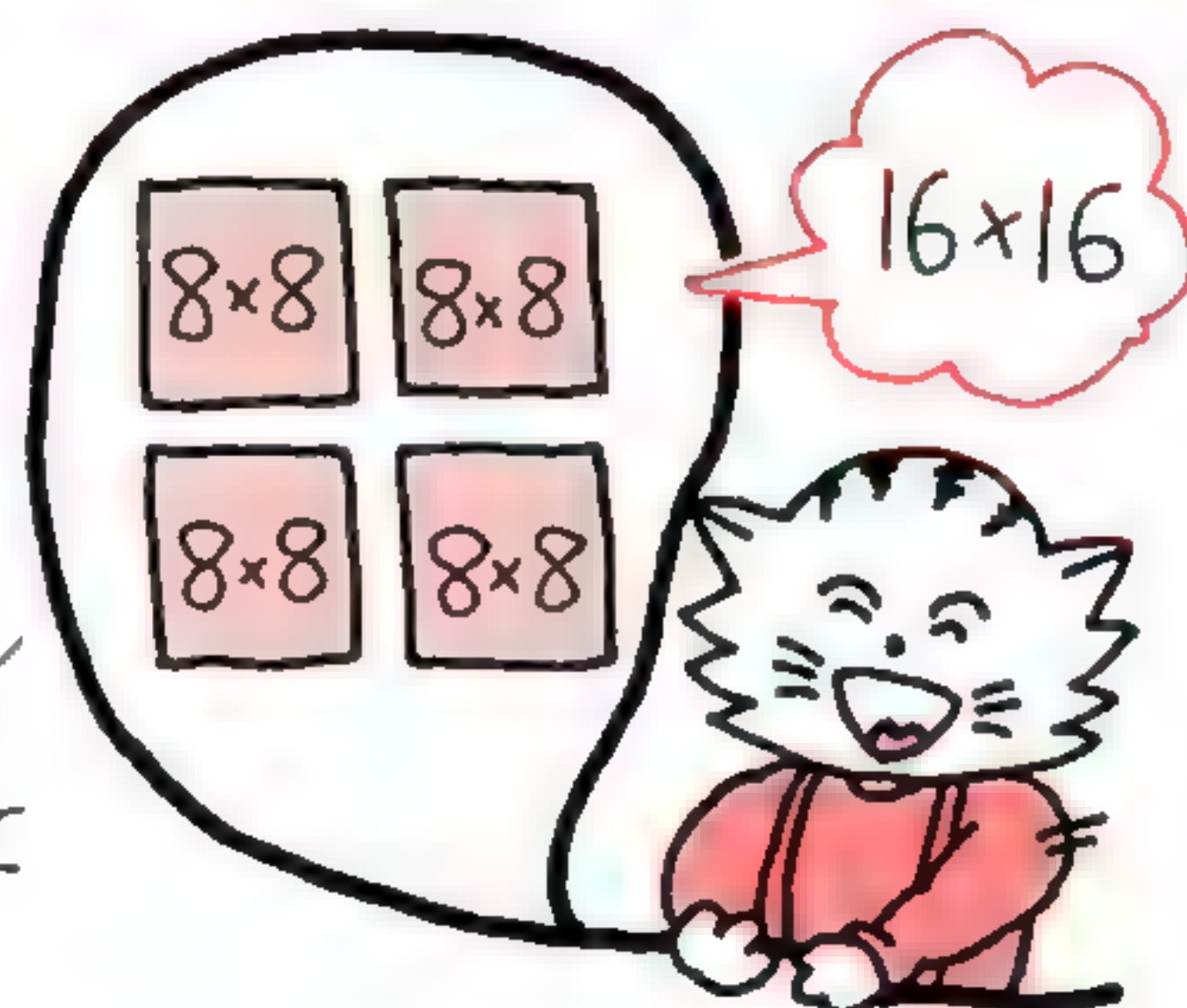
16×16ドットなら、もっと複雑なキャラクタも表示できて、おもしろさもグーンとアップだ

同じディスプレイ画面で、8×8のパターンと16×16のパターンを同時に表示させることはできないよ

●16×16のパターンを定義する

16×16ドットは、8×8ドットの大きなマスが4つ集まったと考えよう。

縦横16個のマス目を左右に2つ、上下に2つに分ければいいんだ。

パターンを定義する順番は、左上が一番先だよ。

これだけのマス目を2進数であらわしていたら、たいへんだ。メモリのむだ使いにもなる。そこで、こんどは10進数でパターンを定義してみよう。

●インベーダーが左から右へ歩くプログラム

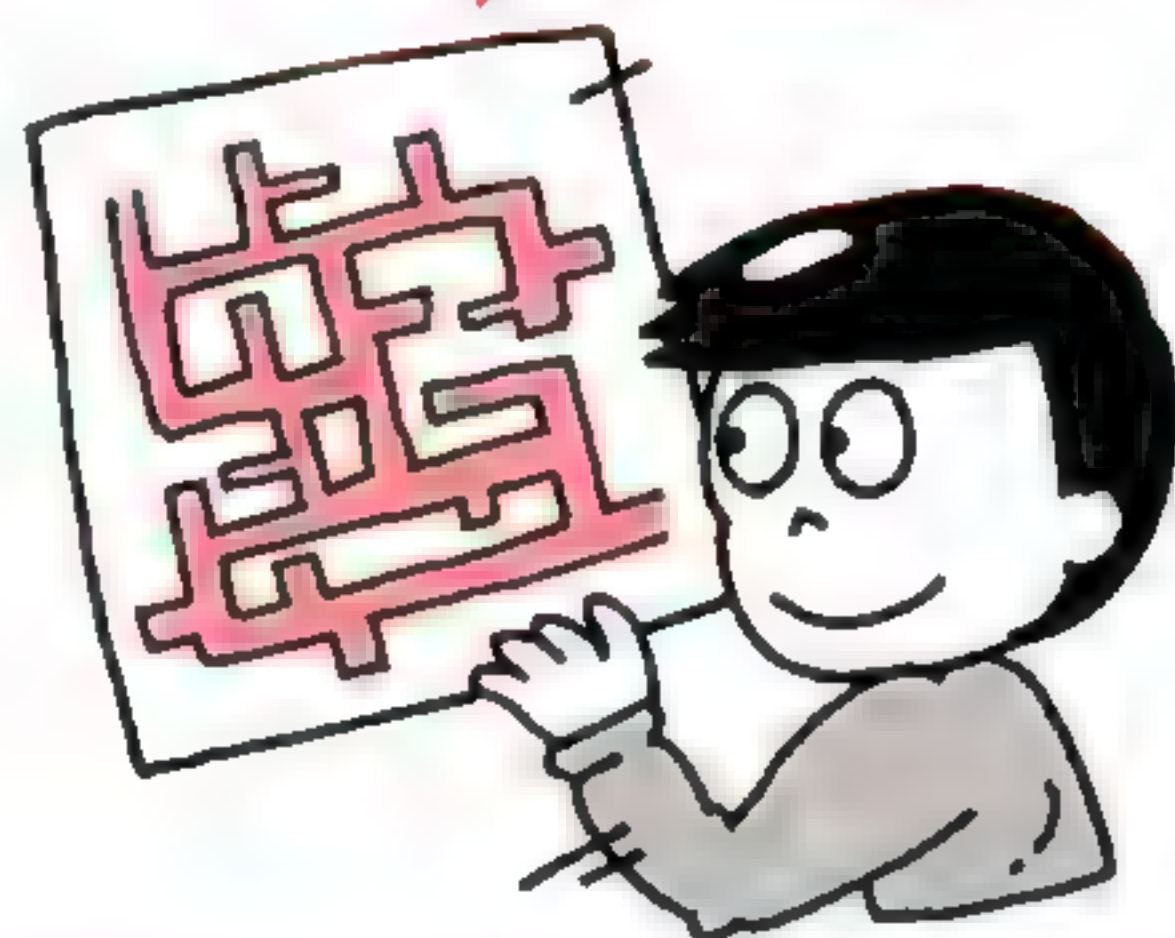
```
10 SCREEN 1,2
20 FOR I=1 TO2
30 SP$=""
40 FOR J=1 TO32
50 READ A:A$=CHR$(A)
60 SP$=SP$+A$
70 NEXT J
80 SPRITE$(I)=SP$
90 NEXTI
100 FOR K=16 TO232 STEP 16
110 PUT SPRITE 1,(K,100),10,1
120 GOSUB 170
130 PUT SPRITE 1,(K+8,100),10,2
140 GOSUB 170
150 NEXT K
160 END
170 PLAY "L20O3c"
180 FOR L=1 TO300:NEXT L
190 RETURN
200 DATA &h3,&h3,&h1f,&h1f,&h3f,&h3f,&h39,&h39
210 DATA &h3f,&h3f,&h6,&h6,&hd,&hd,&h30,&h30
220 DATA &hc0,&hc0,&hf8,&hf8,&hfc,&hfc,&h9c,&h9c
230 DATA &hfc,&hfc,&h60,&h60,&hb0,&hb0,&hc,&hc
240 DATA &H3,&H3,&H1f,&H1f,&H3f,&H3f,&H39,&H39
250 DATA &h3f,&h3f,&he,&he,&h18,&h18,&hc,&hc
260 DATA &hc0,&hc0,&hf8,&hf8,&hfc,&hfc,&h9c,&h9c
270 DATA &hfc,&hfc,&h70,&h70,&h18,&h18,&h30,&h30
```

10進数の値は出せたかな？　1段目から順に、3、3、31、31、63、63、57、57になったかな。白い部分は、たしちゃだめだよ。

上のプログラムは、左ページの2つのパターンを交たいに表示させて、画面を左から右へ歩いているように見せたプログラムだよ。

3次元迷路脱出ゲーム

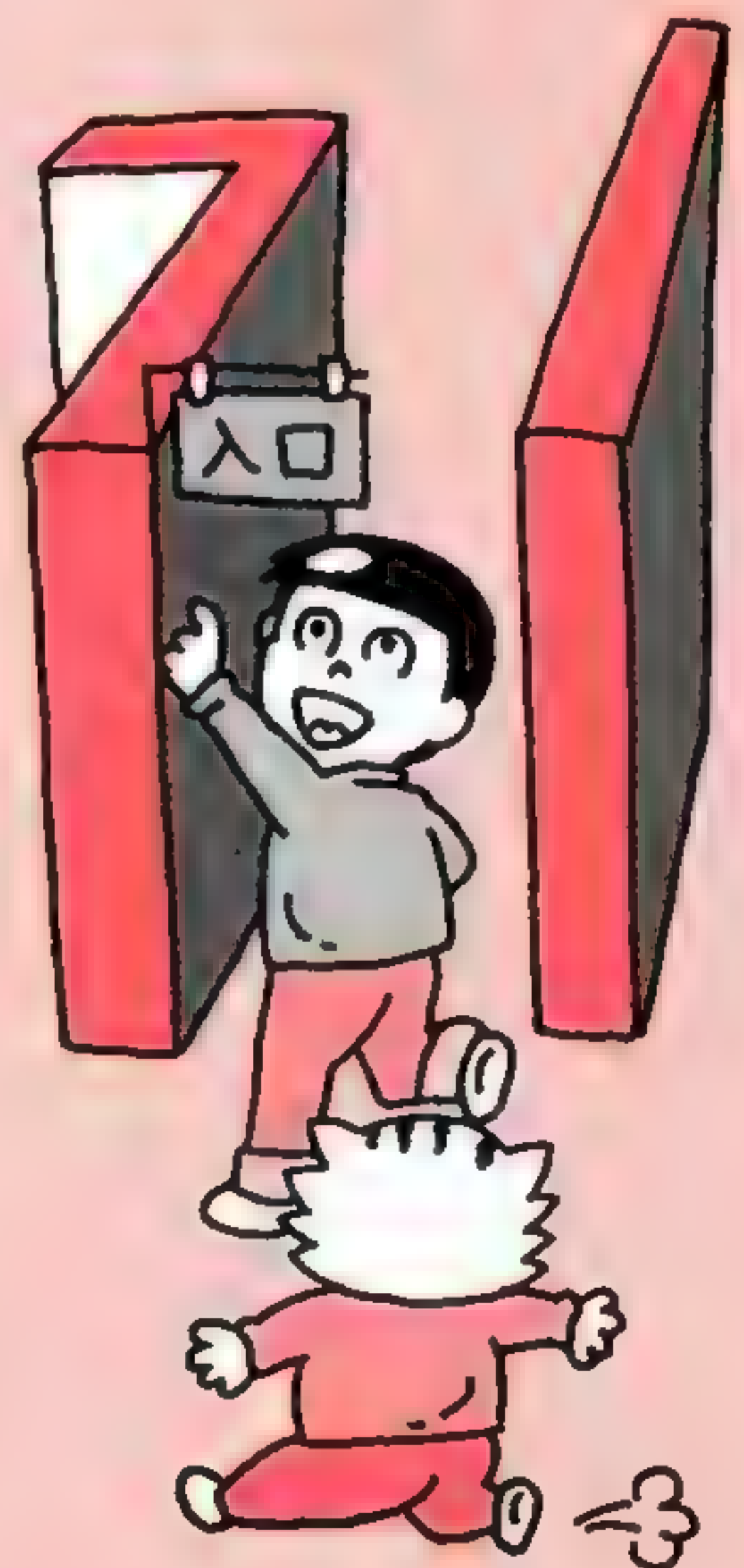
迷路の平面図を頭に入れてから

RUNすると、コンピュータが迷路を自動的に作成する。1分ぐらいたつと、画面に迷路の平面図が登場する。しばらくすると、この図は消えてしまうから、それまでに、どういう順序で進んで行ったら、迷路から脱出できるのかをしっかり頭にたたき込んでおくのだ。

その後が、実際のゲームだ。きみがスタート位置に立ったとしよう。立体図がどんどんきみに迫ってくるから、キー入力をしながらゴールをめざせ。スタートは、LOCATE（18, 18）だ。Yの値が少なくなるにつれて上に、Xの値が少なくなるにつれて左に移動していることになるぞ。

向かっている方向を示す矢印を見ながら、
U……1歩前進
H……90°左へ変更
K……90°右へ変更
M……うしろを向く
どんどんキー入力だ

ゲームプログラム　3次元迷路からの脱出

```
10 DIM M$(22,22),W$(4,2)
20 SCREEN 0
30 CLS
40 WIDTH 40:COLOR 10 ,1
50 LOCATE 10,2:PRINT"*** 3ｼﾞｹﾞﾝ ﾒｲﾛ***"
60 LOCATE 5,5
70 PRINT"ﾘｯﾀｲ ﾋｮｳｼﾞ ｻﾚﾙ ﾒｲﾛ ｦ ﾐﾅｶﾞﾗ,"
```

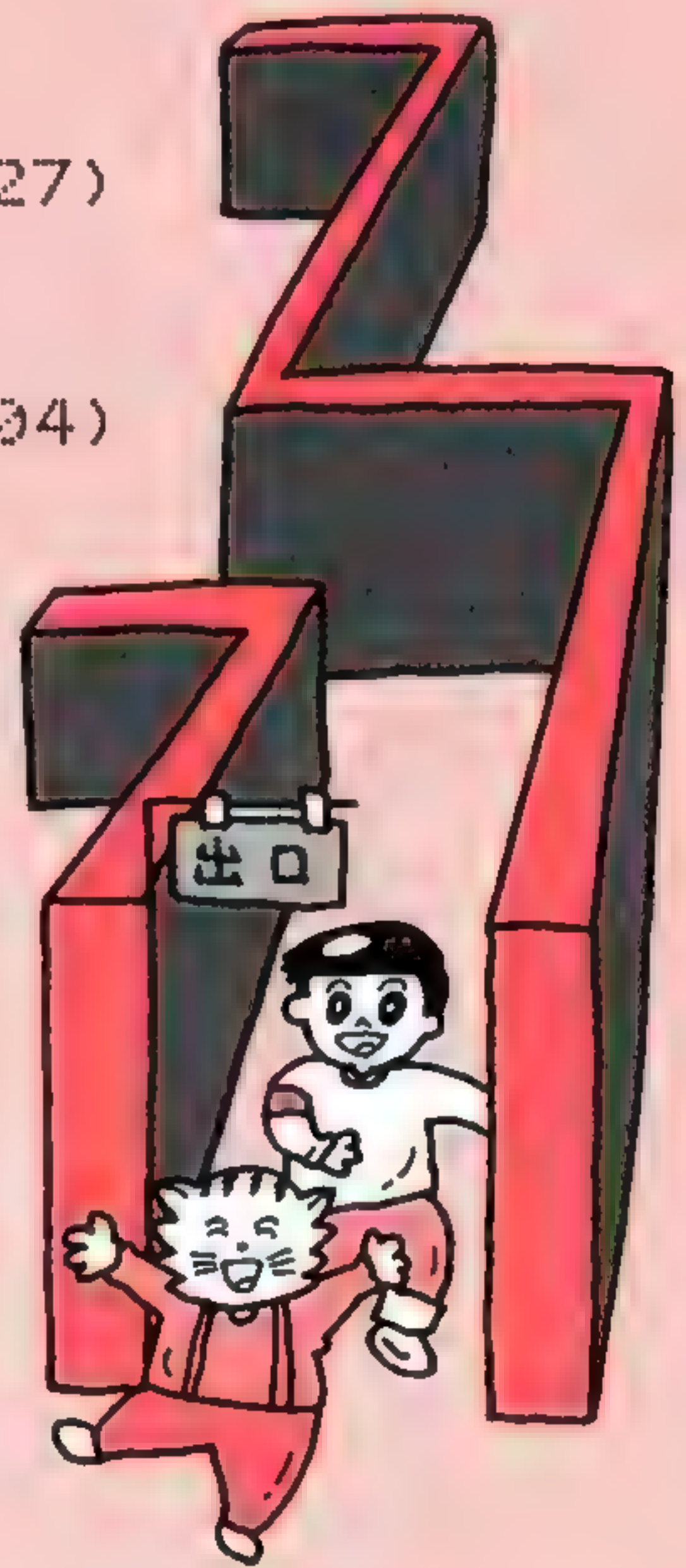
```
80 LOCATE 17,7
90 PRINT"ﾒｲﾛ ｦ ﾇｹﾃ ｸﾀﾞｻｲ ｡"
100 LOCATE 10,10:PRINT"(U)KEY...ﾏ ｴ"
110 LOCATE 10,12:PRINT"(H)KEY...ﾋﾀﾞﾘ  "
120 LOCATE 10,14:PRINT"(K)KEY...ﾐ ｷﾞ"
130 LOCATE 10,16:PRINT"(M)KEY...ｳｼﾛ "
140 LOCATE 10,18
150 LOCATE 14,21:PRINT"GET KEY"
160 K$=INKEY$:IF K$="" THEN 160
170 GOSUB 660
180 GOSUB 1880
190 SCREEN 2:OPEN"GRP:"AS#1
200 X=20:Y=20:HX=0:HY=-1
210 TIME=0
220 GOSUB 310
230 K$=INKEY$:IF K$="" THEN 230
240 IF K$="k" THEN GOSUB 610:GOTO 290
250 IF K$="h" THEN GOSUB 640:GOTO 290
260 IF K$="m" THEN HX=-HX:HY=-HY:GOTO 290
270 IF K$="u" THEN GOSUB 540:GOTO 290
280 GOTO 230
290 IF (X=2)AND(Y=2) THEN 460
300 GOTO 220
310 REM
320 CLS
330 GOSUB 1910
340 GOSUB 2130
350 FOR I=0 TO 4
360 PRESET(20*8,5*8-I*8)
370 FOR J=0 TO 2
380 PRINT#1,W$(I,J);
390 NEXT J,I
400 PRESET(21*8,5*8):PRINT#1,"!"
410 PRESET(17*8,9*8)
420 PRINT#1,"ﾎｳｲ ";:GOSUB 2900
430 PRESET(17*8,10*8)
440 PRINT#1,"ｲﾁ ";X-2;" ,";Y-2
450 RETURN
460 REM
470 GOSUB 310
480 PRESET(17*8,14*8):PRINT#1,"ﾀﾞｯｼｭﾂ!"
```

```
490 PRESET(17*8,16*8):PRINT#1,"TIME ";
500 PRINT #1,TIME
510 PRESET(4*8,20*8):PRINT#1,"ANY KEY"
520 K$=INKEY$:IF K$="" THEN GOTO 520
530 RUN
540 REM
550 XE=X:YE=Y
560 X=X+HX:Y=Y+HY
570 IF M$(X,Y)="◆" THEN X=XE:Y=YE
580 IF M$(X,Y)="♥" THEN X=XE:Y=YE
590 RETURN
600 REM
610 R1=HX*0+HY*-1:R2=HX*1+HY*0
620 HX=R1:HY=R2:RETURN
630 REM
640 R1=HX*0+HY*1:R2=HX*-1+HY*0
650 HX=R1:HY=R2:RETURN
660 REM
670 CLS
680 FOR J=0 TO 22:FOR I=0 TO 20
690 M$(I,J)="◆"
700 NEXT I
710 COLOR 4,15
720 PRINT    "◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆"
730 NEXT J
740 FOR I=0 TO 22
750 M$(I,0)="♥":M$(I,1)="♥"
760 LOCATE I,0:PRINT"♥"
770 LOCATE I,1:PRINT"♥"
780 NEXT I
790 FOR I=0 TO 21
800 M$(I,21)="♥":M$(I,22)="♥"
810 LOCATE I,21:PRINT"♥"
820 LOCATE I,22:PRINT"♥";
830 NEXT I
840 FOR I=0 TO 22
850 M$(0,I)="♥":M$(1,I)="♥"
860 LOCATE 0,I:PRINT"♥♥"
870 NEXT I
880 FOR I=0 TO21
890 M$(21,I)="♥":M$(22,I)="♥"
```

```
900 LOCATE 21,I:PRINT"♥♥"
910 NEXTI
920 FOR I=2 TO 20 STEP 2
930 FOR J=2 TO 20 STEP 2
940 IF M$(I,J)=" " THEN 1530
950 A1$=M$(I-2,J)
960 A2$=M$(I+2,J)
970 A3$=M$(I,J-2)
980 A4$=M$(I,J+2)
990 IF A1$+A2$+A3$+A4$="    " THEN 1530
1000 II=I:JJ=J
1010 M$(II,JJ)=" ":LOCATE II,JJ:PRINT" "
1020 IF M$(II,JJ-1)="♥" THEN 1050
1030 M$(II,JJ-1)=" "
1040 LOCATE II,JJ-1:PRINT" "
1050 IF JJ<>2 THEN 1090
1060 IF M$(II-1,JJ)="♥" THEN 1090
1070 M$(II-1,JJ)=" "
1080 LOCATE II-1,JJ:PRINT" "
1090 ONINT(RND(-TIME)*4)+1GOTO1100,1170,1240,1310
1100 IF M$(II-2,JJ)="♥" THEN 1090
1110 IF M$(II-2,JJ)=" " THEN 1380
1120 M$(II-1,JJ)=" "
1130 LOCATE II-1,JJ:PRINT" "
1140 M$(II-2,JJ)=" "
1150 LOCATE II-2,JJ:PRINT" "
1160 II=II-2:GOTO 1380
1170 IF M$(II+2,JJ)="♥" THEN 1090
1180 IF M$(II+2,JJ)=" " THEN 1380
1190 M$(II+1,JJ)=" "
1200 LOCATE II+1,JJ:PRINT" "
1210 M$(II+2,JJ)=" "
1220 LOCATE II+2,JJ:PRINT" "
1230 II=II+2:GOTO 1380
1240 IF M$(II,JJ-2)="♥" THEN 1090
1250 IF M$(II,JJ-2)=" " THEN 1380
1260 M$(II,JJ-1)=" "
1270 LOCATE II,JJ-1:PRINT" "
1280 M$(II,JJ-2)=" "
1290 LOCATE II,JJ-2:PRINT" "
1300 JJ=JJ-2:GOTO 1380
```

```
1310 IF M$(II,JJ+2)="♥" THEN 1090
1320 IF M$(II,JJ+2)=" " THEN 1380
1330 M$(II,JJ+1)=" "
1340 LOCATE II,JJ+1:PRINT" "
1350 M$(II,JJ+2)=" "
1360 LOCATE II,JJ+2:PRINT" "
1370 JJ=JJ+2:GOTO 1380
1380 B1$=M$(II-2,JJ)
1390 B2$=M$(II+2,JJ)
1400 B3$=M$(II,JJ-2)
1410 B4$=M$(II,JJ+2)
1420 B$=B1$+B2$+B3$+B4$
1430 IF B$="    " THEN 1530
1440 IF B$="♥   " THEN 1530
1450 IF B$=" ♥  " THEN 1530
1460 IF B$="  ♥ " THEN 1530
1470 IF B$="   ♥" THEN 1530
1480 IF B$="♥ ♥ " THEN 1530
1490 IF B$=" ♥♥ " THEN 1530
1500 IF B$="♥  ♥" THEN 1530
1510 IF B$=" ♥ ♥" THEN 1530
1520 GOTO 1090
1530 NEXTJ,I
1540 FOR I=2 TO 20 STEP 2
1550 FOR J=2 TO 20 STEP 2
1560 C1$=M$(I-2,J)
1570 C2$=M$(I+2,J)
1580 C3$=M$(I,J-2)
1590 C4$=M$(I,J+2)
1600 C$=C1$+C2$+C3$+C4$
1610 IF (C$="    ")AND(M$(I,J)="◆")THEN 1640
1620 NEXT J,I
1630 GOTO 1790
1640 M$(I,J)=" ":LOCATE I,J:PRINT" "
1650 ONINT(RND(-1)*4)+1 GOTO 1660,1700,1730,1760
1660 IF M$(I-1,J)="♥" THEN 1640
1670 M$(I-1,J)=" "
1680 LOCATE I-1,J:PRINT" "
1690 GOTO 1620
1700 IF M$(I+1,J)="♥" THEN 1640
1710 M$(I+1,J)=" ":LOCATE I+1,J:PRINT" "
```

```
1720 GOTO 1620
1730 IF M$(I,J-1)=" " THEN 1640
1740 M$(I,J-1)=" ":LOCATE I,J-1:PRINT" "
1750 GOTO 1620
1760 IF M$(I,J+1)="♥" THEN 1640
1770 M$(I,J+1)=" ":LOCATE I,J+1:PRINT" "
1780 GOTO 1620
1790 FOR K=0 TO10
1800 LOCATE 2,1:PRINT"!"
1810 LOCATE 20,21:PRINT"!"
1820 FOR J=0 TO 500:NEXT J
1830 LOCATE 2,1:PRINT"♥"
1840 LOCATE 20,21:PRINT"♥"
1850 FOR J=0 TO 500:NEXT J
1860 NEXT K
1870 RETURN
1880 REM
1890 CLS
1900 RETURN
1910 REM
1920 FOR I=0 TO 4
1930 X1=X+HX*I:Y1=Y+HY*I
1940 IF (X1<0)OR(Y1<0)THEN W$(I,1)=" ":GOTO 1970
1950 IF(X1>22)OR(Y1>22)THENW$(I,1)=" ":GOTO 1970
1960 W$(I,1)=M$(X1,Y1)
1970 NEXT
1980 R1=HX*0+HY*1:R2=HX*-1+HY*0
1990 FOR I=0 TO 4
2000 X1=X+R1+HX*I:Y1=Y+R2+HY*I
2010 IF(X1<0)OR(Y1<0)THENW$(I,0)=" ":GOTO 2040
2020 IF(X1>22)OR(Y1>22)THENW$(I,0)=" ":GOTO 2040
2030 W$(I,0)=M$(X1,Y1)
2040 NEXT
2050 R1=HX*0+HY*-1:R2=HX*1+HY*0
2060 FOR I=0 TO 4
2070 X1=X+R1+HX*I:Y1=Y+R2+HY*I
2080 IF(X1<0)OR(Y1<0)THENW$(I,2)=" ":GOTO 2110
2090 IF(X1>22)OR(Y1>22)THENW$(I,2)=" ":GOTO 2110
2100 W$(I,2)=M$(X1,Y1)
2110 NEXT
2120 RETURN
```

```
2130 REM
2140 FOR I=1 TO 4
2150 IF W$(I,1)="♥" THEN 2180
2160 IF W$(I,1)="♦" THEN 2180
2170 NEXT
2180 ON I GOTO 2280,2250,2220,2190,2330
2190 LINE (71,71)-(80,71):LINE-(80,80)
2200 LINE-(71,80):LINE-(71,71)
2210 GOTO 2330
2220 LINE (64,63)-(87,63)
2230 LINE (64,88)-(87,88)
2240 GOTO 2470
2250 LINE (48,47)-(103,47)
2260 LINE (48,104)-(103,104)
2270 GOTO 2590
2280 LINE (24,23)-(127,23)
2290 LINE(24,128)-(127,128)
2300 LINE (23,24)-(23,127)
2310 LINE (128,24)-(128,128)
2320 GOTO 2710
2330 IF W$(3,0)=" " THEN 2370
2340 LINE (64,64)-(71,71)
2350 LINE (64,87)-(71,80)
2360 GOTO 2400
2370 LINE (64,71)-(71,71)
2380 LINE(64,80)-(71,80)
2390 LINE (71,71)-(71,80)
2400 IF W$(3,2)=" " THEN 2440
2410 LINE (80,71)-(87,64)
2420 LINE (80,80)-(87,87)
2430 GOTO 2470
2440 LINE (80,71)-(87,71)
2450 LINE(80,80)-(87,80)
2460 LINE (80,71)-(80,80)
2470 IF W$(2,0)=" " THEN 2510
2480 LINE (48,48)-(63,63)
2490 LINE -(63,88):LINE -(48,103)
2500 GOTO 2530
2510 LINE (48,63)-(63,63)
2520 LINE -(63,88):LINE -(48,88)
2530 IF W$(2,2)=" " THEN 2570
```

```
2540 LINE (103,48)-(88,63)
2550 LINE -(88,88):LINE-(103,103)
2560 GOTO 2590
2570 LINE (103,63)-(88,63)
2580 LINE-(88,88):LINE-(103,88)
2590 IF W$(1,0)=" " THEN 2630
2600 LINE (24,24)-(47,47)
2610 LINE-(47,104):LINE-(24,127)
2620 GOTO 2650
2630 LINE (24,47)-(47,47)
2640 LINE-(47,104):LINE-(24,104)
2650 IF W$(1,2)=" " THEN 2690
2660 LINE (127,24)-(104,47)
2670 LINE-(104,104):LINE-(127,127)
2680 GOTO 2710
2690 LINE (127,47)-(104,47)
2700 LINE-(104,104):LINE-(127,104)
2710 LINE (23,24)-(23,127)
2720 LINE (128,24)-(128,127)
2730 IF W$(0,0)=" " THEN 2770
2740 LINE (16,16)-(23,23)
2750 LINE (16,135)-(23,128)
2760 GOTO 2790
2770 LINE (16,23)-(23,23)
2780 LINE (16,128)-(23,128)
2790 IF W$(0,2)=" " THEN 2830
2800 LINE (128,23)-(135,16)
2810 LINE (128,128)-(135,135)
2820 GOTO 2850
2830 LINE (128,23)-(135,23)
2840 LINE (128,128)-(135,128)
2850 PRESET(72,72)
2860 IF(X=2)AND(Y<5)AND(HY=-1)THEN PRINT#1,"G"
2870 PRESET(72,72)
2880 IF(X<5)AND(Y=2)AND(HX=-1) THEN PRINT#1,"G"
2890 RETURN
2900 IF (HX=0)AND(HY=1) THEN PRINT#1,"S";
2910 IF (HX=0)AND(HY=-1) THEN PRINT#1,"N";
2920 IF (HX=1)AND(HY=0) THEN PRINT#1,"E";
2930 IF (HX=-1)AND(HY=0) THEN PRINT#1,"W"
2940 RETURN
```

キーボード操作のトレーニング

アルファベット&BASIC

●トレーニングのメニューは4つ

キードードをしっかりと正確に押すこと。これはいちばんたいせつなことだ。いっしょうけんめい入力した長いプログラムも、1字まちがえているだけで、エラーを出して途中で止まってしまう。

そこで、キーボードを徹底的にマスターするプログラムを作ってみた。

何回もRUNして練習すれば、キーボードを速く、正確に打てるようになるはずだ。最後には、キーボードを見ずに打てるくらい上達したいね。

プログラムをRUNさせると、キーボードの絵と4つのトレーニングメニューが出てくる。きみが練習したいと思うメニューのところに、カーソルを使って移動して、RETURN キーを押す。

カーソルは上向きと下向きの2つを使うんだよ。

トレーニングメニューは、次の4項目を用意した。

1．アルファベットの練習

アルファベットのA～Zまでを順番に押す練習だ。画面のキーボードに赤い枠が表示さ

1．アルファベットの練習

2．アルファベットのテスト

れるから、その通りキーを押していく。

2．アルファベットのテスト

はじめにＡ～Ｚまでのアルファベットを押すのに何秒かかるかをテストする。そのあとで、ランダムに出てくるアルファベットを１分間に何回押せるかをテストする。これは、キーボードを見ずに入力する練習にもなる。

3．BASIC命令の入力練習

3．ＢＡＳＩＣ命令の入力練習

BASICの命令が画面に表示されるから、きみはその命令をキーボードから入力する。１のように、画面上のキーに赤い枠が表示される。

4．BASIC命令の入力テスト

4．ＢＡＳＩＣ命令の入力テスト

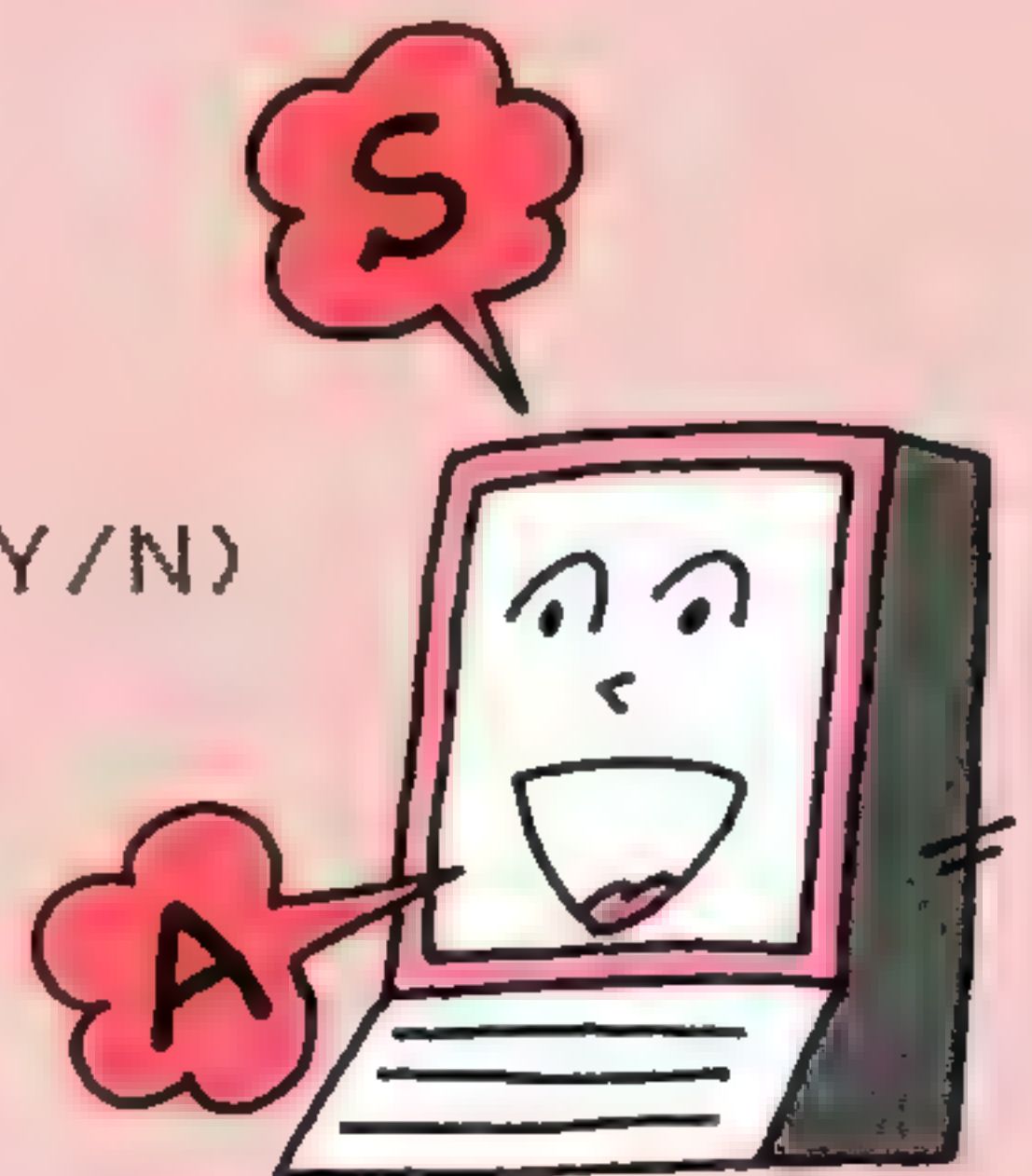
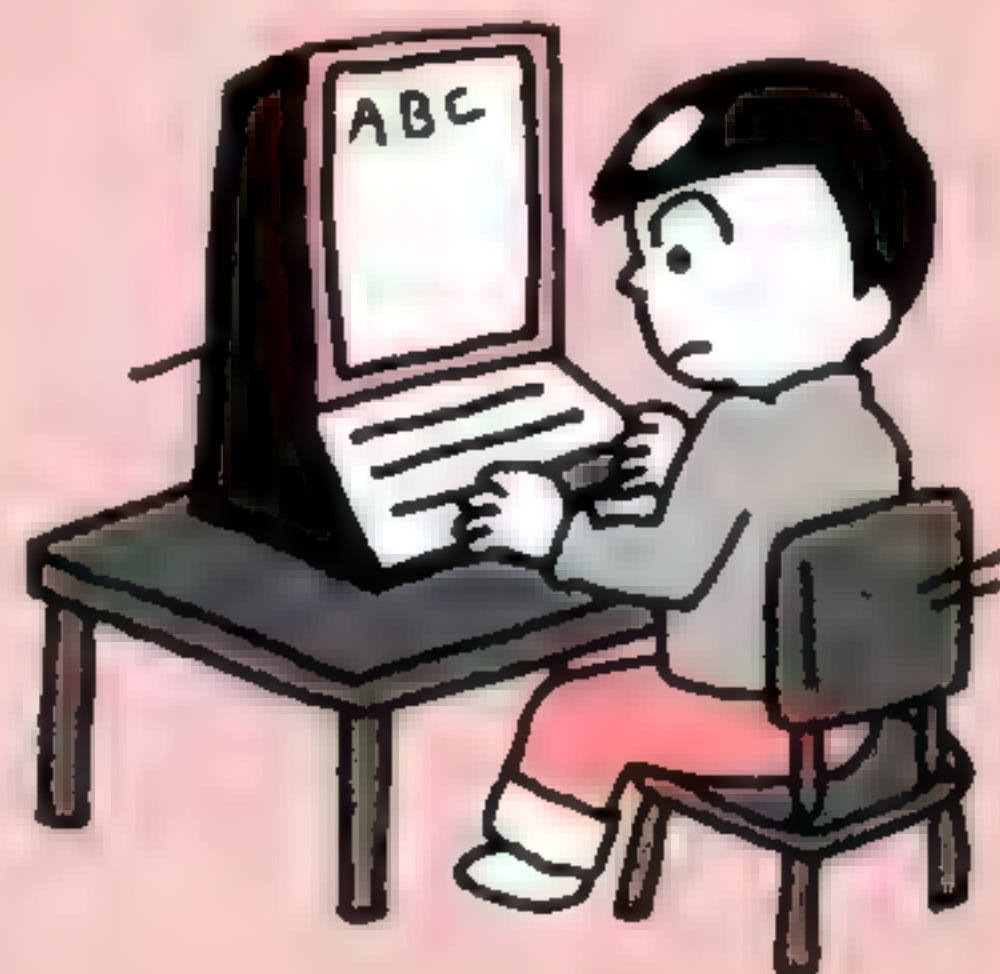
画面に表示されるＢＡＳＩＣの命令をすべて入力するのにどのくらいの時間がかかるかをテストする。

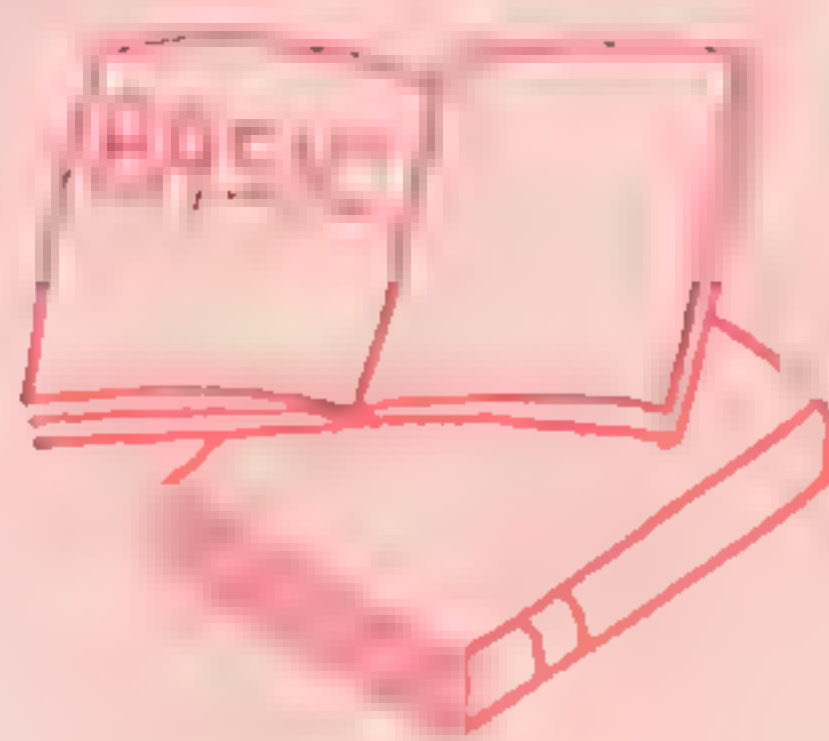
●トレーニングプログラム　キーボードマスター

```
10 DIM AD(26,1),CM$(200),CM(200)
20 COLOR 15,14
30 SCREEN 2,0
40 RESTORE 3580
50 FOR I=0 TO 25
60 READ A,B
70 AD(I,0)=A:AD(I,1)=B
80 NEXT
90 OPEN"grp:"FOR OUTPUT AS#1
100 GOSUB 660
110 SCREEN ,0
```

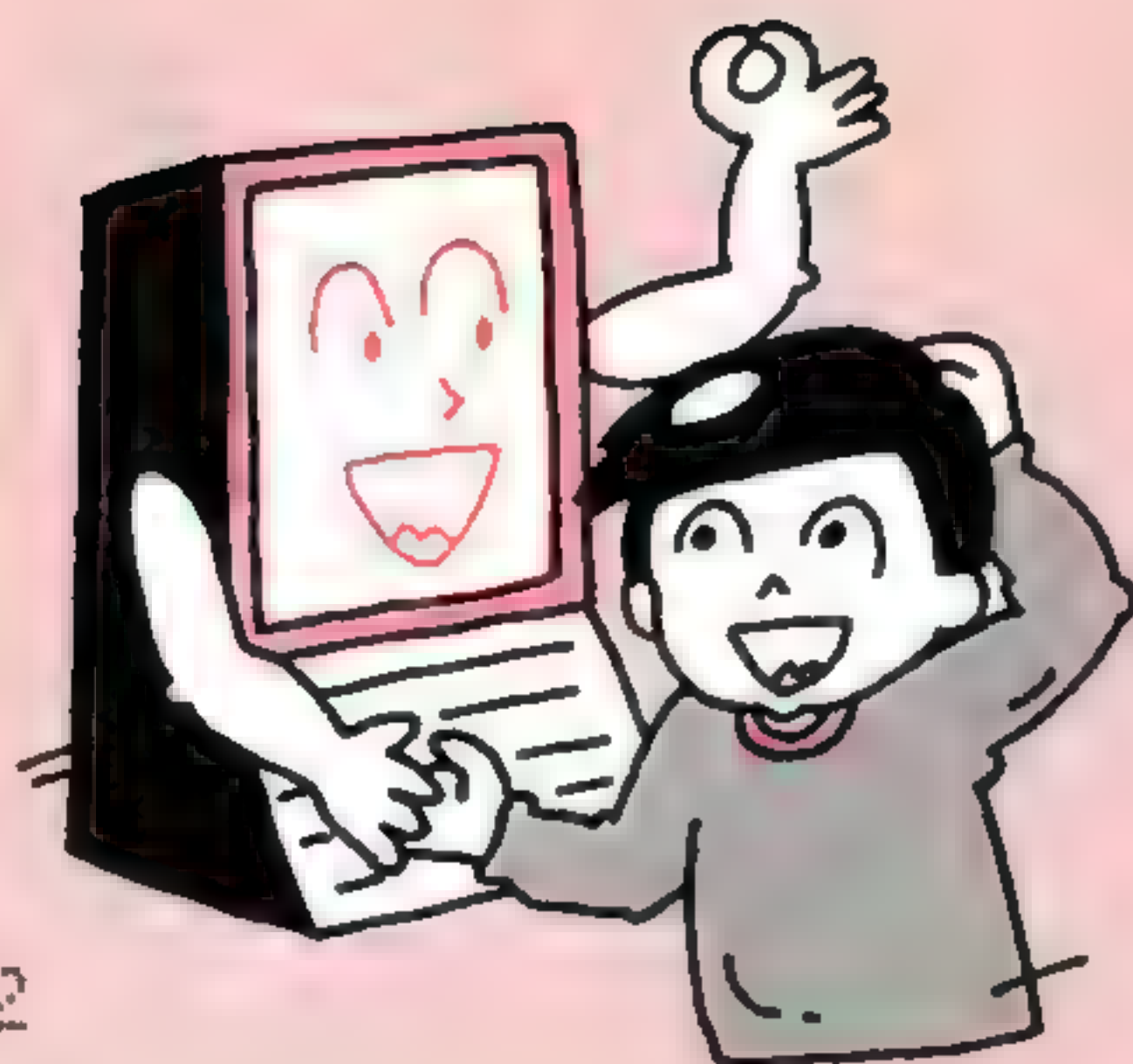
```
120 LINE(12,8)-(248,90),4,BF
130 CA$=""
140 RESTORE 3530
150 FOR I=1 TO8
160 READ CA:CA$=CA$+CHR$(CA)
170 NEXT
180 SPRITE$(1)=CA$
190 FOR I=0 TO 4
200 CIRCLE (60,20+I*16),3,9
210 NEXT
220 COLOR 15,4
230 PRESET (80,16)
240 PRINT#1,"ｱﾙﾌｧﾍﾞｯﾄﾉ ﾚﾝｼｭｳ"
250 PRESET (80,32)
260 PRINT#1,"ｱﾙﾌｧﾍﾞｯﾄﾉ ﾃｽﾄ"
270 PRESET (80,48)
280 PRINT#1,"ｺﾏﾝﾄﾞﾆｭｳﾘｮｸﾉ ﾚﾝｼｭｳ"
290 PRESET (80,64)
300 PRINT#1,"ｺﾏﾝﾄﾞﾆｭｳﾘｮｸﾉ ﾃｽﾄ"
310 PRESET (80,80)
320 PRINT#1,"ｵﾜﾘ"
330 XD=0
340 A$=INKEY$:IF A$<>"" GOTO 340
350 PUT SPRITE 1,(57,17+XD*16),9,1
360 A$=INKEY$
370 IF A$=CHR$(30)ANDXD<>0 THEN XD=XD-1
380 IF A$=CHR$(31)ANDXD<>4 THEN XD=XD+1
390 IF A$=CHR$(13) AND XD=4 THEN END
400 IF A$=CHR$(13) GOTO 420
410 GOTO 350
420 SCREEN ,3
430 RESTORE 3540
440 CA$=""
450 FOR I=1 TO 8
460 READ A:CA$=CA$+CHR$(A)
470 NEXT
480 SPRITE$(2)=CA$
490 RESTORE 3550
500 CA$=""
510 FOR I=1 TO 24
520 READ A:CA$=CA$+CHR$(A)
```

```
530 NEXT
540 SPRITE$(3)=CA$
550 LINE(12,8)-(248,90),4,BF
560 PRESET(24,16)
570 PRINT#1,"ﾊｼﾞﾒﾙﾏｴﾆ CAPSLOCKｷｰﾄ ｶﾅｷｰｶﾞ"
580 PRESET(24,28)
590 PRINT#1,"ｵｻﾚﾃｲﾅｲｺﾄｦ ｶｸﾆﾝｼﾃｸﾀﾞｻｲ｡"
600 PRESET (24,50)
610 PRINT#1,"ｶｸﾆﾝｼﾀﾗ ｽﾍﾟｰｽｷｰｦ ｵｼﾃｸﾀﾞｻｲ｡"
620 A$=INKEY$
630 IF A$<>" " GOTO 620
640 ON XD+1 GOSUB 1010,1340,2150,2720
650 GOTO110
660 'ｷｰﾎﾞｰﾄﾞ ﾋｮｳｼﾞ
670 LINE(12,8)-(248,90),4,BF
680 LINE(12,91)-(248,182),11,BF
690 FOR I=0 TO 12
700 LINE (I*16+24,100)-(I*16+36,112),15,BF
710 NEXT
720 FOR I=0 TO 11
730 LINE (I*16+32,116)-(I*16+44,128),15,BF
740 NEXT
750 FOR I=0 TO 11
760 LINE (I*16+40,132)-(I*16+52,144),15,BF
770 NEXT
780 LINE (24,148)-(44,160),15,BF
790 FOR I=0 TO 11
800 LINE (I*16+48,148)-(I*16+60,160),15,BF
810 NEXT
820 LINE (228,148)-(244,160),15,BF
830 LINE (80,164)-(204,176),15,BF
840 DRAW"BM24,180C7"
850 DRAW"M+4,-16R4M+2,10M+2,-10R4M+3,12"
860 DRAW"R9U2L7U10R18M+3,5M+3,-5R5"
870 DRAW"M-5,8M+5,8L5M-3,-5M-3,5L5M+5,-8M-4,-4"
880 DRAW"L10D3R6D9L15M-2,-10M-2,10"
890 DRAW"L4M-2,-10M-2,10L5"
900 PAINT(25,178),7
910 COLOR 0
920 PRESET(27,102),15
930 PRINT#1,"1 2 3 4 5 6 7 8 9 0 - ^ ¥"
```

```
940 PRESET(35,118),15
950 PRINT#1,"Q W E R T Y U I O P @ ["
960 PRESET(43,134),15
970 PRINT#1,"A S D F G H J K L ; : ]"
980 PRESET(27,150),15
990 PRINT#1,"SH Z X C V B N M , . /   SH"
1000 RETURN
1010 'ｱﾙﾌｧﾍﾞｯﾄﾉ ﾚﾝｼｭｳ
1020 LINE(12,8)-(248,90),4,BF
1030 PRESET (24,16)
1040 PRINT#1,"a ｶﾗ zﾏﾃﾞﾉｱﾙﾌｧﾍﾞｯﾄｦ
1050 PRESET (32,28)
1060 PRINT#1,"ｼﾞｭﾝﾊﾞﾝﾆ ｵｼﾃｸﾀﾞｻｲ｡"
1070 PRESET (24,40)
1080 PRINT#1,"ｶﾞﾒﾝｼﾞｮｳﾉｷｰﾎﾞｰﾄﾞﾆﾊ ｱｶｲﾜｸｶﾞ
1090 PRESET (32,52)
1100 PRINT#1,"ﾋｮｳｼﾞｻﾚﾏｽ｡
1110 PRESET (24,64)
1120 PRINT#1,"ﾊｼﾞﾒﾙﾄｷﾊ ｽﾍﾟｰｽｷｰｦ ｵｼﾃｸﾀﾞｻｲ｡"
1130 A$=INKEY$
1140 IF A$=" "GOTO 1150 ELSE 1130
1150 LINE(20,78)-(234,88),2,BF
1160 FOR I=0 TO 25
1170 PUT SPRITE 2,(AD(I,0),AD(I,1)),8,2
1180 A$=INKEY$:IF A$<>""GOTO 1180
1190 A$=INKEY$:IF A$="" GOTO 1190
1200 IF A$=CHR$(97+I) GOTO 1230
1210 PLAY"c"
1220 GOTO 1190
1230 BEEP
1240 PRESET (24+I*8,80),2
1250 PRINT#1,CHR$(I+97)
1260 NEXT
1270 LINE(12,8)-(248,77),4,BF
1280 PRESET (24,50)
1290 PRINT#1,"ﾓｳｲﾁﾄﾞ ﾔﾘﾏｽｶ ? (Y/N)
1300 A$=INKEY$
1310 IF A$="y" GOTO 1010
1320 IF A$="n" THEN RETURN
1330 GOTO 1300
1340 'ｱﾙﾌｧﾍﾞｯﾄﾉ ﾃｽﾄ
```

```
1350 LINE(12,8)-(248,90),4,BF
1360 PRESET (24,16)
1370 PRINT#1,"コノテストハ a カラ z マテ゛ノモシ゛ヲ
1380 PRESET (32,28)
1390 PRINT#1,"ナンヒ゛ョウテ゛ニュウリョクテ゛キルカヲ
1400 PRESET (32,40)
1410 PRINT#1,"タメステストテ゛ス。
1420 PRESET (24,52)
1430 PRINT#1,"スヘ゜ースキーヲ オスト 3ヒ゛ョウコ゛ニ"
1440 PRESET (32,64)
1450 PRINT#1,"BEEPオンカ゛ナリ テストカ゛ハシ゛マリマス。"
1460 A$=INKEY$
1470 IF A$<>" " GOTO 1460
1480 LINE(12,8)-(248,90),4,BF
1490 TIME=0
1500 IF TIME<180 GOTO 1500
1510 BEEP
1520 LINE(20,28)-(234,40),2,BF
1530 TIME=0
1540 FOR I=0 TO 25
1550 A$=INKEY$:IF A$<>""GOTO 1550
1560 A$=INKEY$:IF A$="" GOTO 1560
1570 IF A$<"a" OR A$>"z" GOTO 1560
1580 AC=ASC(A$)-97
1590 PUT SPRITE 2,(AD(AC,0),AD(AC,1)),8,2
1600 IF A$=CHR$(97+I) GOTO 1630
1610 PLAY"c"
1620 GOTO 1560
1630 BEEP
1640 PRESET (24+I*8,30),2
1650 PRINT#1,CHR$(I+97)
1660 NEXT
1670 TI=TIME
1680 LINE(12,8)-(248,90),4,BF
1690 PRESET (40,16)
1700 PRINT#1,"タタ゛イマノタイムハ";INT(TI/60);"ヒ゛ョウテ゛シタ"
1710 PRESET (24,40)
1720 PRINT#1,"ツキ゛ハ ランタ゛ムニアラワレルアルファヘ゛ットヲ"
1730 PRESET (32,52)
1740 PRINT#1,"1フ゜ンカンニ ナンモシ゛ニュウリョク"
1750 PRESET (32,64)
```

```
1760 PRINT#1,"ﾃﾞｷﾙｶﾗ ﾀﾒｽﾃｽﾄﾃﾞｽ｡"
1770 PRESET (24,76)
1780 PRINT#1,"ｽﾍﾟｰｽｷｰｦ ｵｽﾄﾃｽﾄｶﾞﾊｼﾞﾏﾘﾏｽ｡"
1790 A$=INKEY$
1800 IF A$=" "GOTO 1810 ELSE 1790
1810 LINE(12,8)-(248,90),4,BF
1820 PUT SPRITE 2,(0,209),0,0
1830 LINE (120,40)-(140,60),15,B
1840 FOR J=1 TO 500:NEXT J
1850 SK=0
1860 A=RND(-TIME)
1870 TIME=0
1880 RM=INT(RND(1)*26)+97
1890 PRESET (128,48)
1900 PRINT#1,CHR$(RM-32)
1910 A$=INKEY$
1920 IF TIME>3600 GOTO 2060
1930 IF A$<"a" OR A$>"z" GOTO 1910
1940 AC=ASC(A$)-97
1950 PUT SPRITE 2,(AD(AC,0),AD(AC,1)),8,2
1960 IF A$=CHR$(RM) THEN 1990
1970 PLAY"c"
1980 GOTO 1910
1990 BEEP
2000 SK=SK+1
2010 COLOR 4
2020 PRESET (128,48)
2030 PRINT#1,CHR$(RM-32)
2040 COLOR 15
2050 GOTO 1880
2060 LINE(12,8)-(248,90),4,BF
2070 PRESET (40,16)
2080 PRINT#1,"ﾆｭｳﾘｮｸｼﾀﾓｼﾞｽｳﾊ ";SK;"ﾃﾞｽ"
2090 PRESET (24,40)
2100 PRINT#1,"ﾓｳｲﾁﾄﾞ ﾔﾘﾏｽｶ ? (Y/N)
2110 A$=INKEY$
2120 IF A$="y" GOTO 1340
2130 IF A$="n" THEN RETURN
2140 GOTO 2110
2150 'ｺﾏﾝﾄﾞﾆｭｳﾘｮｸﾉ ﾚﾝｼｭｳ
2160 LINE(12,8)-(248,90),4,BF
```

```
2170 PRESET (24,16)
2180 PRINT#1,"ｶﾞﾒﾝｼﾞｮｳﾆ BASICﾉｺﾏﾝﾄﾞｶﾞ"
2190 PRESET (32,28)
2200 PRINT#1,"ﾃﾞﾃｷﾏｽﾉﾃﾞ ｿﾉｺﾏﾝﾄﾞｦ"
2210 PRESET (32,40)
2220 PRINT#1,"ｷｰﾎﾞｰﾄﾞﾃﾞ ﾆｭｳﾘｮｸｼﾃｸﾀﾞｻｲ｡"
2230 PRESET (24,52)
2240 PRINT#1,"ｶﾞﾒﾝｼﾞｮｳﾉｷｰﾎﾞｰﾄﾞﾆﾊ ｱｶｲﾜｸｶﾞ
2250 PRESET (32,64)
2260 PRINT#1,"ﾋｮｳｼﾞｻﾚﾏｽ｡
2270 PRESET (24,76)
2280 PRINT#1,"ﾊｼﾞﾒﾙﾄｷﾊ ｽﾍﾟｰｽｷｰｦ ｵｼﾃｸﾀﾞｻｲ｡"
2290 A$=INKEY$
2300 IF A$=" "GOTO 2310 ELSE 2290
2310 LINE(12,8)-(248,90),4,BF
2320 PRESET (40,40)
2330 PRINT#1,"ﾀﾀﾞｲﾏ ｺﾏﾝﾄﾞｦﾖﾐｺﾝﾃﾞｲﾏｽﾉﾃﾞ"
2340 PRESET (40,60)
2350 PRINT#1,"ｼﾊﾞﾗｸ ｵﾏﾁｸﾀﾞｻｲ｡"
2360 GOSUB3320
2370 A$=INKEY$:IF A$<>"" GOTO 2370
2380 LINE(12,8)-(248,90),4,BF
2390 LINE(80,28)-(180,40),15,B
2400 FOR I=1 TO KT
2410 LINE(80,58)-(180,70),2,BF
2420 VT$=CM$(CM(I))
2430 PRESET (100,30)
2440 PRINT#1,VT$
2450 SL=LEN(VT$)
2460 FOR J=1 TO SL
2470 MO$=MID$(VT$,J,1)
2480 IF MO$="$" THEN GOSUB 3490:GOTO 2510
2490 FL=ASC(MO$)-97
2500 PUT SPRITE 2,(AD(FL,0),AD(FL,1)),8,2
2510 A$=INKEY$:IF A$="" GOTO2510
2520 PUT SPRITE 3,(0,209),0,0
2530 IF A$=MO$ GOTO 2560
2540 PLAY"c"
2550 GOTO 2510
2560 BEEP
2570 PRESET(92+J*8,60),2
```

```
2580 PRINT#1,MO$
2590 NEXT J
2600 COLOR 4
2610 PRESET (100,30)
2620 PRINT#1,VT$
2630 COLOR 15
2640 NEXT I
2650 LINE(12,8)-(248,77),4,BF
2660 PRESET (40,50)
2670 PRINT#1,"モウイチド ヤリマスカ ? (Y/N)
2680 A$=INKEY$
2690 IF A$="y" GOTO 2150
2700 IF A$="n" THEN RETURN
2710 GOTO 2680
2720 'コマンド ニュウリョクノ テスト
2730 LINE(12,8)-(248,90),4,BF
2740 PRESET (24,16)
2750 PRINT#1,"コレハ ガメンジョウニ デテクル"
2760 PRESET (32,28)
2770 PRINT#1,"BASICノコマンドヲ ゼンブ"
2780 PRESET (32,40)
2790 PRINT#1,"ニュウリョクスルノニ ナンビョウ"
2800 PRESET (32,52)
2810 PRINT#1,"カカルカヲ ハカルテストデス。"
2820 PRESET (24,64)
2830 PRINT#1,"ハジメルトキハ スペースキーヲ オシテクダサイ。"
2840 A$=INKEY$
2850 IF A$=" "GOTO 2860 ELSE 2840
2860 LINE(12,8)-(248,90),4,BF
2870 PRESET (40,40)
2880 PRINT#1,"モンダイヲ サクセイチュウデス。
2890 PRESET (40,60)
2900 PRINT#1,"シバラク オマチクダサイ。"
2910 GOSUB3320
2920 A$=INKEY$:IF A$<>"" GOTO 2920
2930 LINE(12,8)-(248,90),4,BF
2940 TIME=0
2950 LINE(80,28)-(180,40),15,B
2960 FOR I=1 TO KT
2970 LINE(80,58)-(180,70),2,BF
2980 VT$=CM$(CM(I))
```

```
2990 PRESET (100,30)
3000 PRINT#1,VT$
3010 SL=LEN(VT$)
3020 FOR J=1 TO SL
3030 MO$=MID$(VT$,J,1)
3040 A$=INKEY$:IF A$="" GOTO 3040
3050 IF A$="$" THEN GOSUB 3490:GOTO 3090
3060 IF A$<"a" OR A$>"z" THEN 3040
3070 KI=ASC(MO$)-97
3080 PUT SPRITE 2,(AD(KI,0),AD(KI,1)),8,2
3090 IF A$=MO$ GOTO 3120
3100 PLAY"c"
3110 GOTO 3040
3120 BEEP
3130 PRESET (92+J*8,60),2
3140 NEXT J
3150 COLOR 4
3160 PRESET (100,30)
3170 PRINT#1,VT$
3180 COLOR 15
3190 PUT SPRITE 2,(0,209),0,0
3200 PUT SPRITE 3,(0,209),0,0
3210 NEXT I
3220 TI=TIME
3230 LINE(12,8)-(248,90),4,BF
3240 PRESET (24,28)
3250 PRINT#1,"ﾀﾀﾞｲﾏﾉﾀｲﾑﾊ";INT(TI/60);"ﾋﾞｮｳﾃﾞｼﾀ"
3260 PRESET (40,60)
3270 PRINT#1,"ﾓｳｲﾁﾄﾞ ﾔﾘﾏｽｶ ? (Y/N)
3280 A$=INKEY$
3290 IF A$="y" GOTO 2720
3300 IF A$="n" THEN RETURN
3310 GOTO 3280
3320 RESTORE 3650
3330 KT=0
3340 READ C$
3350 IF C$="END" GOTO 3390
3360 KT=KT+1
3370 CM$(KT)=C$
3380 GOTO 3340
3390 FOR I=1 TO KT
```

```
3400 CM(I)=0
3410 NEXT
3420 R=RND(-TIME)
3430 FOR I=1 TO KT
3440 RN=INT(RND(1)*KT+1)
3450 IF CM(RN)<>0 GOTO 3440
3460 CM(RN)=I
3470 NEXT
3480 RETURN
3490 PUT SPRITE 2,(72,99),8,2
3500 PUT SPRITE 3,(20,147),[illegible],3
3510 RETURN
3520 'sprite data
3530 DATA &h18,&h3c,&h7e,&hff,&hff,&h7e,&h3c,&h18
3540 DATA &hfe,&h82,&h82,&h82,&h82,&h82,&hfe,0
3550 DATA &hff,&h80,&h80,&h80,&h80,&h80,&hff,0
3560 DATA 0,0,0,0,0,0,0,0
3570 DATA &hf8,&h8,&h8,&h8,&h8,&h8,&hf8,0
3580 DATA 40,131,112,147,80,147,72,131
3590 DATA 64,115,88,131,104,131,120,131
3600 DATA 144,115,136,131,152,131,168,131
3610 DATA 144,147,128,147,160,115,176,115
3620 DATA 32,115,80,115,56,131,96,115
3630 DATA 128,115,96,147,48,115,64,147
3640 DATA 112,115,48,147
3650 DATA asc,auto,beep,chr$,circle
3660 DATA clear,cload,close,cls,color
3670 DATA cont,csave,data,delete,dim
3680 DATA dim,draw,end,fix,for
3690 DATA next,gosub,goto,if,then
3700 DATA else,inkey$,input,int,left$
3710 DATA len,line,list,locate,lprint
3720 DATA mod,new,open,paint,play
3730 DATA point,preset,print,pset,put
3740 DATA read,rem,renum,restore,return
3750 DATA right$,mid$,rnd,run,screen
3760 DATA sound,space$,stick,sprite$,spc
3770 DATA stop,strig,string$,str$,tab
3780 DATA time,val,width,sprite,to
3790 DATA END
```

●さくいん

パソコンの用語（アイウエオ順）

バイト（byte）　8ビットを1バイトという……………………………(22)

ビット（bit）　情報量の単位。2進数の桁数のこと。1ビットの情報は2進数1桁分だから、0か1である……………………(22)

ファンクションキー（function key）　キーボード上で、プログラムの実行によく使われる命令（機能）を集めた専用キー……(34)

プログラム　コンピュータにさせる仕事の手順を書いたもの…(18)(40)

プログラムモード　いったんメモリへプログラムをぜんぶ記憶させてから実行する方法……………………………………………(40)

フロッピーディスク　プログラムを書き込んだり、また読み出してメモリに入力するための外部記憶装置……………………(19)(35)

変数　データを出し入れする特定の記憶場所。たくさんの変数を区別するために変数名を付ける。数値変数には数値データが、文字変数には、文字データが入る。文字変数には、変数名のうしろに$（ドルマーク）をつける…………………(51)

乱数　不規則な数。サイコロやルーレットもその一種。パソコンでは、RND関数で、乱数を発生させる…………………(98)

MSX・BASIC命令（ABC順）

AUTO（オート）　行番号や行番号の間隔を自動的に発生させる……(59)

CHR$（キャラクタダラー）　キャラクタコードを指定して、文字や記号を求める………………………………………………(159)

CIRCLE（サークル）　位置や色、大きさなどを指定して、画面に円やだ円、扇形などを描く……………………………………(125)

CLOAD（カセットロード）　カセットテープに記録してあるプログラムを読み込んでコンピュータに入力する…………………(35)

MSX
絵でわかるたのしいパソコン入門

1987年5月25日 発 行 ©

著 者 関 口 泰 夫
発行者 富 永 弘 一
印刷所 公和印刷株式会社

発行所 東京都台東区台東2丁目24 株式会社 新星出版社
電話 (831)0743 郵便番号110 振替東京 4-72233

ISBN4-405-06062-2

定価980円

ISBN4-405-06062-2 C2055 ¥980E